目　录

第一章　引　论

1.1　数值计算方法的对象和特点

一、研究的对象

数值计算方法是近代数学的一个重要分支，它是研究各种数学问题的数值解法（近似解法），包括方法的构造和求解过程的理论分析.

数值计算方法又称为计算方法或数值分析，当前是一门与计算机应用密切结合的实用性很强的数学课程.

利用计算机解决科学计算问题一般有以下几个大过程：

实际问题 → 构造数学模型 → 选择数值计算方法 → 程序设计 → 上机计算求出结果

可见，研究怎样通过计算机所能执行的基本运算，求得各类数学问题的数值解，这就是数值计算的根本任务. 更确切地说，数值计算方法是对给定问题的输入数据和所需计算结果之间关系的一种明确的描述.

众所周知，人类的计算能力是计算工具的性能与计算方法效率的总和，因此，计算能力的提高有赖于这两方面. 例如，有人统计过，在 1955～1975 年的 20 年间，计算机的计算速度提高了数千倍，而同一时间内解决一定规模的椭圆型偏微分方程的计算方法效率竟提高了 100 万倍！这说明研究和选择好的数值计算方法对提高计算速度，在某种意义上说比提高计算机速度更重要，因为计算方法研究所付出的代价比改进计算机的硬件设备毕竟要小得多.

随着计算机的发展与普及，继实验方法、理论方法之后，科学

计算已成为科学实践的第三种手段，求解各种数学问题的数值方法已被广泛应用于不仅是自然科学，还包括生命科学、经济科学和社会科学等各领域. 因此，数值计算的内容是相当丰富的，本书仅向大家介绍在微积分、线性代数、常微分方程等基础数学中常用的、行之有效的数值计算方法，并通过典型例子阐明构造算法的基本思想和技巧，引出相应算法的步骤，便于大家更好地应用.

本书的内容可分为三大部分：

（1）数值逼近与数值微积分；

（2）数值代数；

（3）常微分方程的数值解法.

二、主要特点

数值计算方法是以数学问题为研究对象的，它是数学的一个分支，只是它不像纯数学那样只研究数学本身的理论，而着重研究求解的数值计算方法及与此相关的理论，包括方法的收敛性、稳定性及误差分析，还要根据计算机特点研究计算时间最省的计算方法. 有的方法尽管在理论上还不够严格，但通过实际计算，对比分析等手段，被证明是行之有效的也可采用. 因此，数值计算既有纯数学的高度抽象性与严密的科学性的特点，又有应用的广泛性与数值试验的高度技术性的特点，当前已发展成为一门与使用计算机密切结合的实用性很强的数学课程. 它的任务就是提供在计算机上实际可行的、理论可靠的、计算复杂性好的各种数值方法. 除此以外，它还有以下几个基本特点：

1. 采用“构造性”方法

数值方法中许多问题的存在性的证明都是以“构造性”方法为基础. 所谓用构造性的方法证明一个问题的存在性，就是指具体地把这个问题的计算公式构造出来，这种方法不但证明了问题的存在性，而且有了具体的计算公式，就便于编制程序上机计算.

我们用一个简单的例子说明这个特点.

例 1.1　对命题"实系数二次方程

$$x^2+2bx+c=0$$

当 $b^2>c$ 时,有两个实根."

分别给出构造性与非构造性的证明.

证明　(1) 非构造性证明(如用反证法)

令
$$f(x)=x^2+2bx+c$$

若设方程无实根,即 $f(x)$ 没有零点,从而 $f(x)$ 恒不为零. 由于 $f(x)$ 是 x 的连续函数,可知对所有 x,或者 $f(x)$ 是恒正的,或者 $f(x)$ 是恒负的.

根据对 b 和 c 所附加的条件

$$b^2>c$$

有
$$f(-b)=b^2-2b^2+c=c-b^2<0$$

因而 $f(x)$ 必须是恒负的.

另一方面,当 $|x|$ 很大时,$x^2>|2bx+c|$,因此当 $|x|$ 很大时,又有 $f(x)>0$,从而导出矛盾. 这矛盾的由来是假设方程无实根,故假设错误,从而 $f(x)=0$ 至少有一个实根.

其次,若设方程仅有一个实根,则由 $f(x)$ 的连续性必有

① 当 x 很大时,$f(x)>0$; $-x$ 很大时,$f(x)<0$.

或者② 当 x 很大时,$f(x)<0$; $-x$ 很大时,$f(x)>0$.

这和无论 x 取什么符号,只要 $|x|$ 很大,就有 $f(x)>0$ 的事实相矛盾. 所以方程有两个实根. 但上述证明过程并没有提供求根的方法. 称之为"非构造性证明".

(2) 构造性证明

对任意 x,b 和 c,由

$$x^2+2bx+c=x^2+2bx+b^2-b^2+c$$

$$=(x+b)^2+c-b^2$$

因此,当且仅当

$$(x+b)^2+c-b^2=0$$

时,x 是它的根,亦即

$$(x+b)^2=b^2-c$$

将两端开方并注意 $b^2-c>0$,可以推出

当且仅当
$$x+b=\pm\sqrt{b^2-c}$$
时,x 是根,从而

$$x=-b\pm\sqrt{b^2-c} \tag{1.1}$$

这样就证明了方程存在两个实根,并且可以根据式(1.1)具体算出这两个实根,即同时得到计算根的方法.

在本书的各章内容中,许多存在性证明都是运用了构造性的方法.亦即这种证明同时还提供了具体的计算公式,使需要证明其存在的对象通过数值演算过程来完成.

2. 采用“离散化”方法

把求连续变量问题转化为求离散变量问题,称为离散化.

一个连续的数学问题要上机计算,必须进行离散化.离散化可以认为是数值计算中最基本的概念和方法之一.例如把定积分离散成求和,把微分方程离散成差分方程等等.

例 1.2 计算定积分

$$I=\int_a^b f(x)\mathrm{d}x \tag{1.2}$$

解　这是一个连续性的数学问题,直接在计算机上计算有困难.但大家知道,定积分可用复合梯形公式近似计算,即

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{n}\left[\frac{1}{2}(y_0+y_n)+y_1+\cdots+y_{n-1}\right] \tag{1.3}$$

其中 $a=x_0<x_1<\cdots<x_n=b$ 是 n 等分区间 $[a,b]$ 的分点，$y_0,y_1,\cdots,y_n$ 是被积函数在各分点的函数值. 公式(1.3)就是把定积分离散成为求和运算. 用(1.3)的近似公式就可以求出积分(1.2)的近似值.

3. 采用"递推化"方法

所谓递推化，其基本思想就是将一个复杂的计算过程归结为简单过程的多次重复. 由于递推化算法便于编写计算机程序，所以数值计算中的许多数值方法常常采用"递推化"方法.

例 1.3 对给定的 x，计算多项式

$$P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$$

的值.

解 可利用递推公式

$$\begin{cases} u_0=a_n \\ u_k=u_{k-1}x+a_{n-k} \end{cases} \tag{1.4}$$

对 $k=1,2,\cdots,n$ 反复执行式(1.4)，最终得到的 u_n 就是多项式 $P_n(x)$ 的值.

用公式(1.4)计算多项式 $P_n(x)$ 的值不但逻辑结构简单，而且计算量最小. 这个方法就是历史上著名的秦九韶方法.

4. 采用"近似替代"方法

计算机运算必须在有限次停止，所以数值方法常表现为一个无穷过程的截断，把一个无限过程的数学问题，转化为满足一定误差要求的有限步来近似替代.

例 1.4 计算无理数 e 的近似值.

解 根据泰勒公式得

$$e = 1 + 1 + \frac{1}{2!} + \cdots + \frac{1}{n!} + \cdots \tag{1.5}$$

这是一个无限过程，计算机无法实现，只能在式(1.5)中取有限项计算，再估计误差. 若取

$$e \approx 1 + 1 + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

其误差为

$$|R_n| < \frac{e}{(n+1)!} < \frac{3}{(n+1)!}$$

以上这些特点，将在以后各章中进一步体现，了解了这些特点，对学习数值计算方法很有好处.

1.2 误　　差

许多数值方法给出的解答仅仅是所要求的真解的某种近似，因而研究数值方法，必须注重误差分析，分析误差的来源、误差的传播情况以及对计算结果给出合理的误差估计.

一、误差的来源

误差的来源是多方面的，但主要有以下几种：

1. 模型误差

用计算机解决科学计算问题首先要建立数学模型，它是对被描述的实际问题进行抽象、简化而得到的. 因而总是近似的，这就不可避免地要产生误差，我们把这种数学模型的解与实际问题的解之间出现的误差称为模型误差.

2. 观测误差

在数学模型中通常总包含有一些观测数据，如温度、长度、电

压等，这些数据的值一般是由观测或实验得到的，由于观测不可能绝对准确，由此产生的误差称为观测误差.

3. 截断误差（也称为方法误差）

由实际问题建立起来的数学模型，在很多情况下要得到准确解是困难的. 当数学模型不能得到准确解时，通常要用数值方法求它的近似解，例如常把无限的计算过程用有限的计算过程代替，这种模型的准确解和数值方法的准确解之间的误差称为截断误差. 因为截断误差是方法固有的，又称为方法误差.

例 1.5　函数 $f(x)$ 的泰勒（Taylor）展开式

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

其中 ξ 在 x_0 与 x 之间.

若记

$$S_n(x)=\sum_{k=0}^{n}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k$$

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

则

$$f(x)=S_n(x)+R_n(x)$$

若取

$$f(x)\approx S_n(x)$$

则截断误差为 $R_n(x)$，也即用 $f(x)$ 的泰勒展开式的部分和 $S_n(x)$ 来近似函数 $f(x)$，其余项 $R_n(x)$ 就是真值 $f(x)$ 的截断误差.

4. 舍入误差

由于计算机的字长有限，原始数据在计算机上表示会产生误

差，每一次运算又可能产生新的误差，这种误差称为舍入误差或计算误差.

上述几种误差都会影响计算结果的准确性，数值计算中除了研究求解数学问题的数值方法外，还要研究计算结果的误差是否满足精度要求，这就是误差估计问题. 但前两种误差往往不是计算工作者所能独立完成的，数值计算中用不到描述自然现象，也用不到观测测量，而主要研究的是截断误差与舍入误差对计算结果的影响.

重视误差分析和控制误差扩散是十分重要的，没有误差分析的数值计算结果是不可信的.

二、绝对误差、相对误差和有效数字

如何来定义误差？人们常用绝对误差、相对误差或有效数字来说明一个近似值的准确程度.

1. 绝对误差和相对误差

定义 1.1 设 x 为准确值，x^* 为 x 的一个近似值，称 $e_a = x^* - x$ 为近似值 x^* 的绝对误差，简称误差.

必须指出，通常无法得到准确值 x，从而不可能得到 x^* 的绝对误差 e_a 的真值，只能根据测量的情况，估计出误差的绝对值的一个上界 ε_a，即

$$|e_a| = |x^* - x| \leqslant \varepsilon_a$$

这个正数 ε_a 通常叫做近似值 x^* 的绝对误差限. 有了绝对误差限，就可知道真值 x 的范围

$$x^* - \varepsilon_a \leqslant x \leqslant x^* + \varepsilon_a$$

绝对误差的大小，在许多情况下还不能完全刻画一个近似值的准确程度，例如测量 1000 米和 1 米两个长度，若它们的绝对误差都是 1 厘米，显然前者的测量比较准确. 由此可见，决定一个量

的近似值的精确度，除了考虑绝对误差的大小外，还需要考虑该量本身的大小，为此引入相对误差的概念.

定义 1.2 设 x 为准确值，x^* 是近似值，则称

$$e_r = \frac{e_a}{x} = \frac{x^* - x}{x} \qquad (x \neq 0)$$

为近似值 x^* 的相对误差.

在实际计算中，由于真值 x 一般是不知道的，但可以证明，当 e_r 较小时，e_r 中分母 x 可用 x^* 代替，其两者之差是 e_r 的高阶无穷小，故可忽略不计.

相对误差可正可负，它的绝对值上界叫做相对误差限，即如果有正数 ε_r 使

$$|e_r| \leqslant \varepsilon_r$$

则称 ε_r 为 x^* 的相对误差限.

在误差分析中，相对误差比绝对误差更重要.

2. 有效数字

为了可以从近似数的有限位小数表示本身就能知道近似数的精度，我们引入有效数字概念. 大家知道，当 x 有很多位数字时，常按照“四舍五入”原则，取 x 的前几位数字作为 x 的近似值 x^*.

例如：

$$x = \sqrt{2} = 1.414\,213\,562\,37\cdots$$

若只取到小数后四位数字得

$$x^* = 1.4142$$

其误差为 $0.000\,013\,56\cdots$，误差限为 $0.000\,05 = \frac{1}{2} \times 10^{-4}$. 此时称 x^* 准确到小数后第四位，并称由此位算起的前五位数字 14 142 为 x^* 的有效数字.

一般说来,若近似值 x^* 的误差不超过某位数字的半个单位,而从该位数字到 x^* 最左边的那个非零数字(即自左向右看,第一个出现的非零数字)共有 n 位,那么这 n 位数字都称为 x 的有效数字,即

$$x^* = \overbrace{\times\times\cdots\times}^{n\text{位}}$$

误差不超过该位数的半个单位.

自左向右看,第一个非零数.

并称近似值 x^* 具有 n 位有效数字.

例如:

$$|e-2.718| < 0.0005 = \frac{1}{2}\times 0.001$$

故 e 的近似值 2.718 具有四位有效数字.

具有 n 位有效数字的近似值 x^* 也可以用指数形式写成:

$$x^* = \pm a_1.\ a_2a_3\cdots a_n \times 10^m$$

其中,a_1 是 1 到 9 中的一个数字,$a_2,\cdots,a_n$ 是 0 到 9 中的一个数字,m 为整数,且 x^* 的绝对误差为

$$|e_a| = |x - x^*| \leqslant \frac{1}{2}\times 10^{m-n+1}$$

$a_1, a_2, \cdots, a_n$ 是 x^* 的 n 位有效数字.

同一个准确值的不同近似值,有效数字越多,它的绝对误差和相对误差就越小.有了有效数字概念后,下面两个 3.140011 的近似值的写法是有区别的:3.14 和 3.1400,前者是三位有效数字,后者则表示五位有效数字.

1.3　数值计算中应注意的一些问题

由上述讨论可知，误差分析在数值计算中是一个很重要又很复杂的问题. 因为在数值计算中每一步运算都可能产生误差，而一个科学计算问题的解决，往往要经过成千上万次运算，如果每一步运算都分析误差，显然是不可能的，其实也是不必要的. 人们经常这样做，通过对误差的某些传播规律的分析，指出在数值计算中应该注意的一些问题，有助于鉴别计算结果的可靠性并防止误差危害现象的产生. 下面我们给出在数值计算中应该注意的一些问题.

1. 要使用数值稳定的算法

所谓算法，就是给定一些数据，按着某种规定的次序进行计算的一个运算序列. 它是一个近似的计算过程，我们选择一个算法，主要要求它的计算结果能达到给定的精确度. 一般而言，在计算过程中初始数据的误差和计算中产生的舍入误差总是存在的，而数值解是逐步求出的，前一步数值解的误差必然要影响到后一步数值解. 我们把运算过程中舍入误差不增长的算法称为数值稳定的，否则是数值不稳定的. 只有稳定的数值方法才可能给出可靠的计算结果，不稳定的数值方法毫无实用价值. 下面举一个数值计算的例子：

例 1.6　求

$$I_n = \int_0^1 \frac{x^n}{x+5}\mathrm{d}x,\ (n = 0,1,2,\cdots,8) \tag{1.6}$$

的值.

解　由于

$$I_n + 5I_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5}\mathrm{d}x = \int_0^1 x^{n-1}\mathrm{d}x = \frac{1}{n}$$

初值

$$I_0 = \int_0^1 \frac{1}{x+5}\mathrm{d}x = \ln 6 - \ln 5 = \ln 1.2$$

于是可建立递推公式：

$$\begin{cases} I_0 = \ln 1.2 \\ I_n = \dfrac{1}{n} - 5I_{n-1}，(n = 1,2,\cdots,8) \end{cases} \tag{1.7}$$

若取 $$I_0 = \ln 1.2 \approx 0.182$$

按(1.7)式就可以逐步算得

$$I_1 = 1 - 5I_0 \approx 0.090,$$

$$I_2 = \frac{1}{2} - 5I_1 \approx 0.050,$$

$$I_3 = \frac{1}{3} - 5I_2 \approx 0.083,$$

$$I_4 = \frac{1}{4} - 5I_3 \approx -0.165,$$

$$\cdots\cdots\cdots\cdots$$

因为在$[0,1]$上被积函数$\dfrac{x^n}{x+5} \geqslant 0$ （仅当 $x=0$ 时为零）,所以,I_n $(n=0,1,2,\cdots,8)$ 是恒正的.

但在上述计算结果中,I_4 的近似值却是负的,这个结果显然是错的. 为什么会这样呢 ?这就是误差传播所引起的危害. 由递推公式(1.7)可看出,I_{n-1} 的误差是扩大了 5 倍后传给 I_n,因而初值 I_0 的误差对以后各步计算结果的影响,随着 n 的增大愈来愈严重,这就造成 I_4 的计算结果严重失真,显然,这是一个不稳定的算法.

如果改变计算公式,先取一个 I_n 的近似值,用下面的公式

(1.8)倒过来计算 I_{n-1}，I_{n-2},…,即

$$I_{k-1}=\frac{1}{5k}-\frac{1}{5}I_k,\ (k=n,n-1,\cdots,1) \tag{1.8}$$

情况就不同了.我们发现 I_k 的误差减小到$\frac{1}{5}$后传给 I_{k-1},因而初值的误差对以后各步的计算结果的影响是随着 n 的增大而愈来愈小.

初值 I_n 可以这样确定,利用估计式

$$\frac{1}{6(n+1)}<I_n<\frac{1}{5(n+1)}$$

并取
$$I_9\approx\frac{1}{2}\left(\frac{1}{60}+\frac{1}{50}\right)\approx 0.018$$

按公式(1.8)可逐次求得

$$\begin{aligned}
&I_8\approx 0.019, && I_7\approx 0.021,\\
&I_6\approx 0.024, && I_5\approx 0.028,\\
&I_4\approx 0.034, && I_3\approx 0.043,\\
&I_2\approx 0.058, && I_1\approx 0.088,\\
&I_0\approx 0.182.
\end{aligned}$$

显然,这样算出的值比较符合实际,这个算法具有数值稳定性.

上述事实说明,对于同一数学问题,选用的算法不同,效果也大不相同.实际应用中应选用数值稳定的算法,尽量避免使用不稳定的算法.

2. 要避免两个相近的数相减

在数值计算中,两个相近的数作减法运算时有效数字常会损失.例如,求

$$y=\sqrt{x+1}-\sqrt{x} \tag{1.9}$$

之值，当 $x=1000$，取 4 位有效数字计算得

$$\sqrt{x+1}=31.64, \quad \sqrt{x}=31.62$$

两者直接相减得

$$y=0.02$$

这个结果只有一位有效数字，损失了三位有效数字，从而绝对误差和相对误差都变得很大，严重影响计算结果的精度. 这说明必须尽量避免出现这种运算，遇到这种运算时，最好是改变算法，防止这种情形的出现.

例如，若把公式(1.9)处理成

$$y=\sqrt{x+1}-\sqrt{x}=\frac{1}{\sqrt{x+1}+\sqrt{x}}$$

就可以避免两相近数相减引起的有效数字损失，而得到较精确的结果.

类似地，由

$$\ln x-\ln y=\ln \frac{x}{y}$$

$$\sin(x+\varepsilon)-\sin x=2\cos\left(x+\frac{\varepsilon}{2}\right)\sin \frac{\varepsilon}{2}, \text{（当 } \varepsilon \text{ 很小时）}$$

当 x 和 y 很接近时，采用等号右边的算法，有效数字就不损失.

一般地，当 $f(x) \approx f(x^*)$ 时，可用泰勒展开

$$f(x)-f(x^*)=f'(x^*)(x-x^*)+\frac{f''(x^*)}{2!}(x-x^*)^2+\cdots$$

取右端的有限项近似左端. 若无法改变算法，直接计算时就要多保留几位有效数字.

3. 要避免除数的绝对值远小于被除数的绝对值

由误差的传播规律，可得两近似值之商 $\frac{x_1}{x_2}$ 的绝对误差估计

式：

$$e_a\left(\frac{x_1}{x_2}\right)\approx\frac{|x_1|}{|x_2|^2}e_a(x_2)+\frac{1}{|x_2|}e_a(x_1)$$

若 $|x_2|<<|x_1|$，则 $|x_1|/|x_2|^2>>1$，这表明，当 $|x_2|$ 相对地太小时，商的绝对误差可能很大，因此不宜把绝对值太小的数作除数.

4. 要防止大数"吃掉"小数的现象

在数值计算中参加运算的数的数量级有时相差很大，而计算机计算，做加减法时要"对阶"，当绝对值相差很大的两个数进行加、减运算时，绝对值较小的那个数往往被另一个数"吃掉"而不能发挥其作用，甚至会严重影响计算结果的准确性，所以要采取相应的措施，以保证计算结果的准确性. 我们通过具体算例来说明这种情况.

例 1.7　在五位十进制计算机上，计算

$$A=51\,234+\sum_{i=1}^{1\,000}\delta_i,\quad \text{其中}\quad \delta_i=0.9$$

解　先把参加运算的数写成规格化形式

$$51\,234=0.512\,34\times10^5$$

由于在计算机中两数相加时，要先对阶，即把两数都写成绝对值小于 1 而阶码相同的数. 因 $\delta_i=0.9$，对阶时 $\delta_i=0.000\,009\times10^5$，由于计算机只能表示五位小数，所以

$$\begin{aligned}A&=0.512\,34\times10^5+0.000\,009\times10^5+\cdots+0.000\,009\times10^5\\&\triangleq0.512\,34\times10^5\text{（符号}\triangleq\text{表示机器中运算）}\end{aligned}$$

这一结果显然不可靠，这是由于对阶时，δ_i 被当作是 0，从而出现了大数"51 234"吃掉了小数 δ_i 的结果.

如果计算时先把数量级相同的 1 000 个 δ_i 相加，最后再加 51 234，就不会出现大数"吃掉"小数的现象. 这时

$$\sum_{i=1}^{1000}\delta_i=0.9\times 10^3$$

于是

$$A=0.51234\times 10^5+0.00900\times 10^5=52134$$

所以在数值计算中，应先分析计算方案的数量量级，编程序时加以合理安排，使重要的物理量不至于在计算过程中被“吃掉”.

5. 要简化计算步骤，减少运算次数

简化计算步骤非常重要，它直接影响着计算的速度和误差的积累.

例如，计算多项式

$$P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0=\sum_{k=0}^{n}a_kx^k$$

的值，若直接用上面的公式逐项求和运算，计算第 k 项 a_kx^k，需要 k 次乘法，因此，总共需要作$\frac{1}{2}n(n+1)$次乘法和 n 次加法，才能得到一个值，但如果将公式改写成下面的形式：

$$P_n(x)=x(x\cdots(x(a_nx+a_{n-1})+a_{n-2})+\cdots+a_1)+a_0$$

即用秦九韶算法，则只需要 n 次乘法和 n 次加法就可以了. 从上例可以看出简化公式的重要性.

最后指出，任何数值方法只有运用它在计算机上很快地算出可靠结果时，才显示出它的实用价值. 许多事例说明，如果方法选用不当，计算机的利用率就得不到充分发挥. 怎样评价一个数值方法的好坏呢？一般而言，应考虑以下几点：

(1) 计算量小；

(2) 算出的数值解精度高；

(3) 在计算过程中，占用计算机的存贮单元和工作单元少. 然而，上述各点常不能同时兼备，有时往往是相互制约的. 实际计算时，应根据给定的精度要求及计算机的速度、容量等条件，选用相对较好的数值计算方法，以便得出较精确的结果.

学习指导

一、基本要求与重点

1. 了解数值计算方法研究的对象和主要特点.

2. 注重数值计算中的误差分析和误差的传播,并对计算结果给出合理的误差估计.

3. 重视数值计算中应注意的一些问题,学会选用相对较好的数值计算方法.

本章重点:误差及其传播,数值计算中应注意的一些问题.

二、例题分析与解答

例1 设 $x^* = \pm a_1 a_2 a_3 \cdots a_n \times 10^m \neq 0$ 是 x 的具有 n 位有效数字的近似值(其中,a_1 是 1 到 9 中的一个数字,$a_2, \cdots, a_n$ 是 0 到 9 中的一个数字,m 为整数),且有

(1) x^* 的相对误差限 $\dfrac{|x-x^*|}{|x^*|} \leqslant \dfrac{1}{2a_1} \times 10^{1-n}$;

(2) 如果 x^* 的相对误差限

$$\frac{|x-x^*|}{|x^*|} \leqslant \frac{1}{2(a_1+1)} \times 10^{1-n},$$

则 x^* 至少具有 n 位有效数字.

解 由 $x^* = \pm a_1 a_2 a_3 \cdots a_n \times 10^m$,可得

$$a_1 \times 10^m \leqslant |x^*| \leqslant (a_1+1) \times 10^m, \quad |x-x^*| \leqslant \frac{1}{2} \times 10^{m-n+1},$$

所以当 x^* 有 n 位有效数字时,其相对误差

$$\frac{|x-x^*|}{|x^*|} \leqslant \frac{\frac{1}{2} \times 10^{m-n+1}}{a_1 \times 10^m} = \frac{1}{2a_1} \times 10^{1-n} \tag{1}$$

即式(1)得证(其中,$a_1 \neq 0$,是 x^* 的第一位有效数字).

且式(1)也说明,只要知道近似值 x^* 的有效位数 n 和第一个非零数字 a_1,就能写出它的相对误差限.

反之,由
$$|x-x^*|=|x^*|\frac{|x-x^*|}{|x^*|}$$

因为
$$|x^*|\leqslant(a_1+1)\times10^m,$$

$$\frac{|x-x^*|}{|x^*|}\leqslant\frac{1}{2(a_1+1)}\times10^{1-n} \tag{2}$$

所以$|x-x^*|\leqslant(a_1+1)\times10^m\times\frac{1}{2(a_1+1)}\times10^{1-n}=\frac{1}{2}\times10^{m-n+1}$,

即说明 x^* 至少有 n 位有效数字,也即式(2)得证.

例如,若用 $x^*=2.72$ 作为 e 的近似值,则因为 $|e-x^*|=0.001\,718\,171\,6\cdots<0.002<\frac{1}{2}\times10^{-2}$,所以,$x^*=2.72$ 是 e 的具有 3 位有效数字的近似值. 且此处 $a_1=2$,$n=3$,从而由式(1)可知其相对误差限是

$$\frac{|e-x^*|}{|x^*|}\leqslant\frac{1}{2a_1}\times10^{1-3}=\frac{1}{2\times2}\times10^{-2}=\frac{1}{4}\times10^{-2}.$$

作为例 1 的应用,下面再举几个例子:

例 2 已知近似数 x^* 有两位有效数字,试求其相对误差限.

解 此题给出的已知条件是:$n=2$,但并没有给出近似数 x^* 的第一位有效数字 a_1,遇到这种情况时,可按第一位有效数字出现的最不利的情况估计,即令 $a_1=1$,则由公式得

$$\frac{|x-x^*|}{|x^*|}\leqslant\frac{1}{2a_1}\times10^{1-n}=\frac{1}{2\times1}\times10^{1-2}=5\%$$

即近似数 x^* 的相对误差限为 5%.

例 3 已知近似数 x^* 的相对误差限为 0.3%,问 x^* 至少有几

位有效数字？

解 设 x^* 有 n 位有效数字，但由于 x^* 的第一个有效数字 a_1 没有给出，而 a_1 一定是 1 到 9 中的一个数字，又因为

$$0.3\%=\frac{3}{1000}<\frac{1}{2\times10^2}=\frac{1}{2\times(9+1)}\times10^{-1}$$

令 $1-n=-1$，则 $n=2$，则由例 1 的式(2)可知 x^* 至少具有 2 位有效数字.

例 4 为使$\sqrt{70}$的近似数的相对误差小于 0.1%，问查开方表时，要取几位有效数字？

解 设查表时取 n 位有效数字，则由公式

$$\frac{|x-x^*|}{|x^*|}\leqslant\frac{1}{2a_1}\times10^{1-n},$$

注意到 $8\leqslant\sqrt{70}\leqslant9$，可取 $a_1=8$，因此，为使$\sqrt{70}$的近似数的相对误差小于 0.1%，只需要 $n=3$，就有 $\frac{1}{2a_1}\times10^{1-n}=\frac{1}{2\times8}\times10^{1-3}<0.1\%$，即查开方表时，$\sqrt{70}\approx8.37$，取三位有效数字.

例 5 求二次方程 $x^2+(\alpha+\beta)x+10^9=0$ 的根，这里，$\alpha=-10^9$，$\beta=-1$.

解 由于 $x^2+(\alpha+\beta)x+10^9=x^2+(-10^9-1)x+10^9=(x-10^9)(x-1)$，所以，方程的两个根分别为

$$x_1=10^9,\quad x_2=1.$$

但如果应用一般二次方程 $ax^2+bx+c=0$ 的求根公式：

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

由于当遇到 $b^2\gg4|ac|$ 的情形时，有 $|b|\approx\sqrt{b^2-4ac}$，则用上述公式求出的两个根中，总有一个因用了两个相近的近似数相减而严重

不可靠．如本例若在能将规格化的数表示到小数点后8位的计算机上进行计算，则 $-b=-(\alpha+\beta)=10^9+1=0.1\times10^{10}+0.0000000001\times10^{10}$，由于第二项最后两位数“01”在机器上表示不出来，故 β 这个量在上式的计算中不起作用，即在计算机运算时，$-b=-\alpha=10^9$．

通过类似的分析可得

$$\sqrt{b^2-4ac}\approx|b|=10^9$$

所以，求得的两个根分别为

$$x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}\approx\frac{10^9+10^9}{2}=10^9,$$

$$x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}\approx\frac{10^9-10^9}{2}=0,$$

显然，根 x_2 是严重失真的．

为了求得可靠的结果，可以利用根与系数的关系式：$x_1x_2=\frac{c}{a}$，在计算机上采用如下公式：

$$\begin{cases}x_1=\dfrac{-b-\mathrm{sign}(b)\sqrt{b^2-4ac}}{2a}\\[2ex] x_2=\dfrac{c}{ax_1}\end{cases}$$

其中，$\mathrm{sign}(b)$ 是 b 的符号函数，当 $b>0$ 时 $\mathrm{sign}(b)=1$；当 $b<0$ 时，$\mathrm{sign}(b)=-1$．显然，上述求根公式避免了相近数相减的可能性．

例6　对命题“每一个二次代数方程至多有两个不同的实根”分别给出构造性证明与非构造性证明．

证

(1) 构造性证明

设 $ax^2+bx+c=0$ 为给定的二次代数方程，其中 a,b,c 为实

数且 $a\neq0$，则由代数知道只有三种情况：

1）当 $b^2>4ac$ 时，方程有两个不同的实根：

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a};$$

2）当 $b^2=4ac$ 时，方程有两个相同的实根：$x_{1,2}=-\dfrac{b}{2a}$；

3）当 $b^2<4ac$ 时，方程没有实根．

综上所述，命题得证．

（2）非构造性证明

用反证法，假设方程有三个不同的实根 x_1,x_2,x_3，则将其代入方程，应有

$$\begin{cases}ax_1^2+bx_1+c=0\\ax_2^2+bx_2+c=0\\ax_3^2+bx_3+c=0\end{cases}$$

由于行列式

$$\begin{vmatrix}x_1^2 & x_1 & 1\\x_2^2 & x_2 & 1\\x_3^2 & x_3 & 1\end{vmatrix}=(x_1-x_2)(x_1-x_3)(x_2-x_3)\neq0$$

所以得 $a=b=c=0$，这与题设 $a\neq0$ 矛盾．从而命题得证．

非构造性证明并未给出求根的方法；而构造性证明是通过具体构造出求根公式来证明命题的．

例 7 计算积分 $I_n=\int_0^1 x^n e^{x-1}dx,\quad n=1,2,\cdots$

解 利用分部积分法，有

$$\int_0^1 x^n e^{x-1}dx=\int_0^1 x^n de^{x-1}=x^n e^{x-1}\Big|_0^1-\int_0^1 e^{x-1}nx^{n-1}dx$$

$$= 1 - n\int_0^1 x^{n-1} e^{x-1} dx$$

可得递推公式:

$$I_n = 1 - nI_{n-1} \quad (n = 1,2,\cdots) I_0 = \int_0^1 x^0 e^{x-1} dx = 1 - \frac{1}{e} \tag{1.10}$$

利用公式(1.10)计算 I_n,由于初值 I_0 有误差,不妨设求 I_0 的近似值 I_0^* 时有大小为 ε 的误差,即

$$I_0^* = I_0 + \varepsilon$$

则由递推公式(1.10)得

$$I_1^* = 1 - I_0^* = 1 - I_0 - \varepsilon = I_1 - \varepsilon$$

$$I_2^* = 1 - 2I_1^* = 1 - 2I_1 + 2\varepsilon = I_2 + 2\varepsilon$$

$$I_3^* = 1 - 3I_2^* = 1 - 3I_2 - 6\varepsilon = I_3 - 3!\varepsilon$$

$$I_4^* = 1 - 4I_3^* = 1 - 4I_3 + 4 \times 3!\varepsilon = I_4 + 4!\varepsilon$$

$$\vdots$$

$$I_n^* = I_n + (-1)^n n!\varepsilon$$

显然初始数据的误差 ε 是按 $n!$ 的倍数增长的,误差传播得非常快,例如当 $n=13$ 时,$13! \approx 6.277 \times 10^9$,则 $|I_{13}^* - I_{13}| = 13! \cdot \varepsilon$,即求 I_{13} 时已把初始误差 ε 扩大了 60 亿倍以上,从而 I_{13}^* 的误差已把 I_{13} 的真值淹没掉了,计算结果完全失真.

但如果将递推公式(1.10)改成

$$I_{n-1} = \frac{1}{n}(1 - I_n) \quad (n = k, k-1, \cdots, 3, 2)$$

于是,在从后往前计算时,I_n 的误差减少为原来的 $\frac{1}{n}$,所以,若取 n

足够大，误差逐步减小，显然，计算的结果是可靠的．所以，在构造或选择一种计算公式时，必须考虑到它的数值稳定性问题，数值不稳定的算法是不能使用的．

习 题 一

1．对下述命题分别给出构造性与非构造性证明：

至少有两个实数 x，满足

$$2\cos(2\mathrm{Arccos}x)=1.$$

2．当 N 充分大时，如何计算

$$I=\int_{N}^{N+1}\frac{1}{1+x^2}\mathrm{d}x \quad ?$$

3．计算 $a=(\sqrt{2}-1)^6$，取 $\sqrt{2}\approx 1.4$，采用下列等式计算：

(1) $\dfrac{1}{(\sqrt{2}+1)^6}$；

(2) $99-70\sqrt{2}$；

(3) $(3-2\sqrt{2})^3$；

(4) $\dfrac{1}{(3+2\sqrt{2})^3}$.

问哪一个得到的结果最好？

4．试导出两数近似值之和、差、积的绝对误差与相对误差估计式．

5．数列 $\{x_n\}$ 满足递推公式

$$x_n=10x_{n-1}-1,(n=1,2,\cdots)$$

若取 $x_0=\sqrt{2}\approx 1.41$（三位有效数字），问按上述递推公式，从 x_0 计算到 x_{10} 时误差有多大？这个计算过程稳定吗？

6．求方程 $x^2-56x+1=0$ 的两个根，使它至少具有四位有效数字（已知 $\sqrt{783}\approx 27.982$）．

第二章　插值与逼近

为了计算函数值或分析函数的性态，必须首先产生函数的可计算的近似式. 函数的插值与逼近就是研究用简单函数为各种离散数据建立连续的数学模型；为各种非有理函数提供好的逼近，使它们既能达到精确度要求又使计算量尽可能小. 插值与逼近的理论是数值计算的最基本的内容，对这两方面的研究，无论是对实际应用还是对数值计算领域本身都是极其重要的.

2.1　插值的基本概念

我们知道，许多实际问题都可用函数 $y=f(x)$ 来表示变量间的某种内在规律的数量关系，但是，在工程技术与科学研究中，有时对一个函数 $f(x)$ 只能通过实验或观测等手段得到它在某区间 $[a,b]$ 上的有限个不同点 x_i 上的函数值 $y_i=f(x_i)$，$(i=0,1,2,\cdots,n)$，也即只知道一张函数表，却没有明确的函数表达式；有时，虽然函数有明确的解析表达式，但由于形式复杂，不便于计算和使用，通常宁可先在某些点上取值后造一张函数表. 为了从给定的函数表进一步研究函数的性质，人们往往希望作出一个既能反映函数的特性，又便于计算的简单函数 $P(x)$ 去近似代替 $f(x)$，并且还要求 $P(x)$ 满足条件：$P(x_i)=y_i$，$(i=0,1,2,\cdots,n)$，则称这类问题为插值问题，称 $P(x)$ 为插值函数.

插值函数 $P(x)$ 的选择，取决于使用上的需要，常用的插值函数类有代数多项式类、分段多项式类、有理函数类及三角函数类等，其中，以多项式或分段多项式最便于计算和使用，当选择 $P(x)$ 为代数多项式时，就称为代数插值. 本章重点讨论代数插值.

定义 2.1 设函数 $y=f(x)$在区间$[a,b]$上有定义，且已知它在 $n+1$ 个互异点 $a\leqslant x_0<x_1<\cdots<x_n\leqslant b$ 上的函数值 $y_0,y_1,\cdots,y_n$，若存在一个次数不超过 n 次的多项式

$$P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n,（其中\ a_i\ 为实数）$$

满足条件
$$P_n(x_i)=y_i,(i=0,1,2,\cdots,n) \tag{2.1}$$

则称 $P_n(x)$为函数 $f(x)$的 n 次代数插值多项式.

相应的插值问题称为 n 次代数插值问题，求插值函数 $P_n(x)$的方法称为插值法. 点 $x_0,x_1,\cdots,x_n$ 称为插值节点或简称为节点，函数 $f(x)$称为被插值函数，条件(2.1)称为插值条件，$[a,b]$称为插值区间，点 x 称为插值点. 插值点在插值区间内的叫做内插，否则叫做外插.

代数插值有明确的几何意义，就是通过平面上给定的$(n+1)$个互异点(x_i,y_i)，$(i=0,1,2,\cdots,n)$作一条次数不超过 n 次的代数曲线 $y=P_n(x)$近似地表示曲线 $y=f(x)$. 见图 2-1.

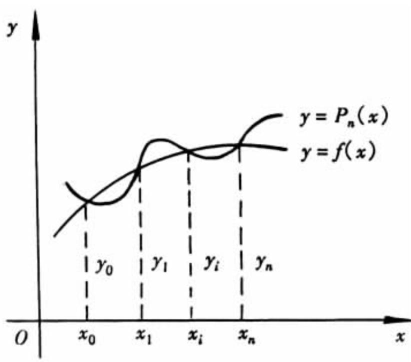


图 2-1　代数插值的几何意义

现在要问，满足插值条件式(2.1)的多项式 $P_n(x)$是否存在？如果存在，则有多少？（回答是肯定的）

定理 2.1 n 次代数插值问题的解是存在且唯一的.

证明 设 n 次多项式 $P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$ 是函数 $y=f(x)$ 在 $[a,b]$ 上的 $n+1$ 个互异的节点 $x_i(i=0,1,2,\cdots,n)$ 上的插值多项式，则求 $P_n(x)$ 的问题就可归结为求它的系数 a_i $(i=0,1,2,\cdots,n)$. 由插值条件：$P_n(x_i)=y_i,(i=0,1,2,\cdots,n)$ 可得关于系数 $a_0,a_1,\cdots,a_n$ 为未知元的 $n+1$ 阶线性代数方程组

$$\begin{cases} a_0+a_1x_0+a_2x_0^2+\cdots+a_nx_0^n=y_0 \\ a_0+a_1x_1+a_2x_1^2+\cdots+a_nx_1^n=y_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_0+a_1x_n+a_2x_n^2+\cdots+a_nx_n^n=y_n \end{cases} \tag{2.2}$$

其系数行列式是范德蒙德（Vandermonde）行列式

$$V(x_0,x_1,\cdots,x_n)=\begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix}=\prod_{i=1}^{n}\prod_{j=0}^{i-1}(x_i-x_j) \tag{2.3}$$

因为 x_i 互不相同，所以式(2.3)不为零，根据解线性方程组的克莱姆（Cramer）法则，方程组的解 a_i 存在且唯一，从而 $P_n(x)$ 被唯一确定，这就证得了 n 次代数插值问题的解是存在且唯一的.

唯一性说明，不论用什么方法来构造，也不论用什么形式来表示插值多项式 $P_n(x)$，只要满足同样的插值条件式(2.1)，其结果都是互相恒等的.

插值问题的关键，是怎样具体去求插值多项式. 事实上，从上述定理的证明过程已提供了一种构造插值多项式 $P_n(x)$ 的方法，即可以通过求方程组(2.2)的解 $a_0,a_1,\cdots,a_n$ 得到. 但这样做当 n 较大时，不但计算工作量大，且因不能获得简明实用的表达式而给理论研究与应用带来不便，因而利用解方程组(2.2)的方法去构造插值多项式是不可取的. 然而，既然已证得了 $P_n(x)$ 是唯一存在的，那么就可以用其他行之有效的方法来构造实用的插值多项式.

下面分别介绍几种常用的构造代数插值多项式的方法.

2.2 拉格朗日(Lagrange)插值

一、基本插值多项式

为了构造满足插值条件 $P_n(x_i)=y_i(i=0,1,2,\cdots,n)$ 的便于应用的插值多项式 $P_n(x)$，我们首先考虑一个最简单的插值问题：求一个 n 次插值多项式 $l_k(x)$，使它在各节点 $x_i(i=0,1,2,\cdots,n)$ 上的值为

$$l_k(x_i)=\begin{cases}1, & i=k\\ 0, & i\neq k\end{cases}, \quad (i,k=0,1,2,\cdots,n) \tag{2.4}$$

显然，这个问题容易解决. 事实上，由条件 $l_k(x_i)=0,(i\neq k)$ 知 $x_0, x_1, \cdots, x_{k-1}, x_{k+1}, \cdots, x_n$ 都是 $l_k(x)$ 的零点，因此可设

$$l_k(x)=A_k(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)$$

其中，A_k 为待定系数，由条件 $l_k(x_k)=1$ 且 x_i 互异可确定系数 A_k，于是得

$$l_k(x)=\frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

$$=\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j} \tag{2.5}$$

由式(2.5)所表示的 n 次代数多项式 $l_k(x),(k=0,1,2,\cdots,n)$ 称为以 $x_0, x_1, \cdots, x_n$ 为节点的 n 次基本插值多项式或 n 次拉格朗日插值基函数.

二、拉格朗日插值多项式

以 $n+1$ 个 n 次基本插值多项式 $l_k(x)$，$(k=0,1,\cdots,n)$ 为基

础，就能直接写出满足插值条件(2.1)的 n 次代数插值多项式.

定理 2.2 n 次代数插值问题的解可表示为

$$P_n(x)=l_0(x)y_0+l_1(x)y_1+\cdots+l_n(x)y_n=\sum_{k=0}^{n}l_k(x)y_k \tag{2.6}$$

证明 由于 $l_k(x),(k=0,1,2,\cdots,n)$ 都是 n 次多项式，所以它们的和 $P_n(x)=\sum_{k=0}^{n}l_k(x)y_k$ 是次数不超过 n 次的多项式，且由

$$l_k(x_i)=\delta_{ki}=\begin{cases}1, & i=k\\ 0, & i\neq k\end{cases}$$

得 $P_n(x_i)=\sum_{k=0}^{n}l_k(x_i)y_k=\sum_{k=0}^{n}\delta_{ki}y_k=y_i,(i=0,1,2,\cdots,n)$

所以公式(2.6)就是满足要求的 n 次代数插值多项式.

我们称形如式(2.6)的插值多项式为拉格朗日插值多项式，并记为 $L_n(x)$，即

$$L_n(x)=\sum_{k=0}^{n}l_k(x)y_k \tag{2.7}$$

由定理 2.1 知，式(2.7)必定与解方程组(2.2)所确定的插值多项式相同. 因而，它只是插值多项式的另一种表示形式. 上述的构造插值多项式的方法称为基函数法.

例如，当 $n=1$ 时，拉格朗日插值多项式(2.7)为

$$L_1(x)=l_0(x)y_0+l_1(x)y_1$$

即
$$L_1(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1 \tag{2.8}$$

用 $L_1(x)$ 近似代替 $f(x)$ 称为线性插值，公式(2.8)称为线性插值多项式或一次插值多项式. 当 $n=2$ 时，拉格朗日插值多项式(2.7)为

$$L_2(x)=l_0(x)y_0+l_1(x)y_1+l_2(x)y_2$$

即
$$L_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2 \tag{2.9}$$

用 $L_2(x)$ 近似代替 $f(x)$ 称为二次插值或抛物线插值，称式(2.9)为二次插值多项式.

从式(2.5)看到，拉格朗日插值基函数 $l_k(x)$，$(k=0,1,2,\cdots,n)$ 仅与节点有关，与被插值函数 $f(x)$ 无关. 从式(2.7)则可看到，拉格朗日插值多项式仅由数据 $(x_i,y_i)(i=0,1,\cdots,n)$ 确定，而与数据的排列次序无关.

三、插值余项

在 $[a,b]$ 上若用 $L_n(x)$ 近似代替 $f(x)$，那么，在节点 x_i 上，恒有 $L_n(x_i)=f(x_i)$，$(i=0,1,2,\cdots,n)$，而在其他的点 $x\in[a,b]$ 上，$L_n(x)$ 与 $f(x)$ 一般是不相等的，即存在误差(见图 2-1).

若记 $R_n(x)=f(x)-L_n(x)$，则 $R_n(x)$ 就是用 $L_n(x)$ 近似代替 $f(x)$ 时的截断误差. 我们称 $R_n(x)$ 为拉格朗日插值多项式的插值余项，它是 $[a,b]$ 上的函数. 关于插值余项有下面的余项估计定理.

定理 2.3 若 $f(x)$ 在 $[a,b]$ 上存在 $n+1$ 阶导数，则对任意给定的 $x\in[a,b]$，插值余项为

$$R_n(x)=f(x)-L_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j) \tag{2.10}$$

其中，$\xi\in(a,b)$ 且依赖于 x.

证明 当 $x=x_i$ 时，式(2.10)的两边均为零，定理的结论显然成立. 现设 $x\neq x_i$，显然由插值条件：$L_n(x_i)=f(x_i)$，可得 $R_n(x_i)=f(x_i)-L_n(x_i)=0$，$(i=0,1,2,\cdots,n)$，所以节点 x_0，x_1，$\cdots$，x_n

都是$R_n(x)$的零点，故可设

$$R_n(x)=k(x)\prod_{j=0}^{n}(x-x_j) \tag{2.11}$$

其中，$k(x)$是待定的函数.

为了求得$k(x)$，我们暂把x看成区间$[a,b]$上一个任意固定而异于$x_i(i=0,1,2,\cdots,n)$的点，并作辅助函数

$$g(t)=f(t)-L_n(t)-k(x)(t-x_0)(t-x_1)\cdots(t-x_n)$$

则易证$g(t)$具有下列性质：

(1) $g(t)$在$[a,b]$上存在$n+1$阶导数，且

$$g^{(n+1)}(t)=f^{(n+1)}(t)-k(x)(n+1)! \tag{2.12}$$

(2) $g(t)$在$[a,b]$上至少有$n+2$个互异的零点：$t=x,x_0,x_1,\cdots,x_n$. 这是因为由插值条件：$f(x_i)=L_n(x_i)$知，当$t=x_i$时，有

$$g(x_i)=0,\ (i=0,1,2,\cdots,n)$$

又由余项定义知，当$t=x$时，有

$$\begin{aligned}g(x)&=f(x)-L_n(x)-k(x)\prod_{j=0}^{n}(x-x_j)\\&=R_n(x)-R_n(x)=0,\ (x\neq x_i)\end{aligned}$$

由罗尔(Rolle)定理可知，$g'(t)$在$g(t)$的每两个零点之间至少有一个零点，故$g'(t)$在(a,b)内至少有$(n+1)$个互异的零点$\xi_i^{(1)}$使

$$g'(\xi_i^{(1)})=0,\quad (i=0,1,2,\cdots,n)$$

对$g'(t)$再用罗尔定理，知$g''(t)$在(a,b)内至少有n个互异的零点$\xi_i^{(2)}$使

$$g''(\xi_i^{(2)})=0,\quad (i=0,1,2,\cdots,n-1)$$

依次类推，反复应用罗尔定理，最后可知至少存在一点$\xi\in(a,b)$，

使得 $g^{(n+1)}(\xi)=0$，于是由式(2.12)得

$$f^{(n+1)}(\xi)-k(x)(n+1)!=0$$

从而得到

$$k(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}$$

将它代入式(2.11)，即得余项的表达式：

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j),\quad \xi\in(a,b).$$ 证毕.

以上证明方法似乎很巧妙，但它却是分析一些误差公式常用的方法.

在具体应用时，因式(2.10)中的 ξ 一般无法具体确定，所以计算 $R_n(x)$ 仍然有困难. 但如果 $f^{(n+1)}(x)$ 在 $[a,b]$ 上有界，即存在常数 $M>0$，使 $|f^{(n+1)}(x)|\leqslant M$，$(x\in[a,b])$ 成立，则有余项估计式

$$|R_n(x)|\leqslant\frac{M}{(n+1)!}\left|\prod_{j=0}^{n}(x-x_j)\right|$$

利用这个估计式可以在计算插值结果前就能估计出它的截断误差.

例 2.1 已知函数 $y=\ln x$ 的函数表如下：

x	10	11	12	13	14
$\ln x$	2.3026	2.3979	2.4849	2.5649	2.6391

试分别用线性插值和抛物线插值求 ln11.75 的近似值，并估计截断误差.

解 在插值计算中，为了减少截断误差，在选择插值节点时，应尽量选取与插值点 x 距离较近的一些节点. 本题中 $x=11.75$ 介于节点 11 与 12 之间，故作线性插值时应取节点 $x_0=11$，$x_1=12$，用线性插值公式(2.8)有

$$L_1(x)=\frac{x-12}{11-12}\times 2.3979+\frac{x-11}{12-11}\times 2.4849=0.087x+1.4409$$

将 $x=11.75$ 代入，即得

$$\ln 11.75\approx L_1(11.75)\approx 2.4632$$

再用插值余项公式(2.10)可得线性插值的余项

$$R_1(x)=\frac{f''(\xi)}{2!}(x-x_0)(x-x_1),\quad (\xi\in[x_0,x_1])$$

本题中 $f(x)=\ln x$，而 $(\ln x)''=-\frac{1}{x^2}$，ξ 在 11 与 12 之间，故有

$$|f''(\xi)|=\frac{1}{\xi^2}\leqslant\frac{1}{11^2}$$

于是用线性插值计算 ln11.75 的截断误差为

$$|R_1(11.75)|\leqslant\frac{1}{2!\,11^2}|(11.75-11)(11.75-12)|<0.0008$$

类似地，在抛物线插值时取节点 $x_0=11$，$x_1=12$，$x_2=13$，所得 ln11.75 的近似值与截断误差分别为

$$\ln 11.75\approx L_2(11.75)=\frac{(11.75-12)(11.75-13)}{(11-12)(11-13)}\times 2.3979$$

$$+\frac{(11.75-11)(11.75-13)}{(12-11)(12-13)}\times 2.4849$$

$$+\frac{(11.75-11)(11.75-12)}{(13-11)(13-12)}\times 2.5649$$

$$\approx 2.4638$$

$$|R_2(11.75)|\leqslant\frac{2}{3!\,11^3}|(11.75-11)(11.75-12)(11.75-13)|$$

$$<0.00006.$$

查对数表得 $\ln 11.75\approx 2.46385.$

由此可见,抛物线插值的精度较线性插值好.

另外,为了便于上机计算,也常将 $L_n(x)$ 的表达式(2.7)写为

$$L_n(x)=\sum_{k=0}^{n}\left[\prod_{\substack{j=0\\j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}\right]y_k$$

编制程序时,可用二重循环来完成 $L_n(x)$ 值的计算,即先固定 k,令 j 从 0 到 $n(j\neq k)$ 作乘积;然后对 k 求和,即得 $L_n(x)$ 的值.

2.3 牛顿(Newton)插值

拉格朗日插值公式含义直观,构造容易,很适合在计算机上使用.但它的一个主要缺点是,如果要再增加一个节点,拉格朗日插值多项式除了要增加一项外,原来的每一项都要改变,则整个计算工作必须从头开始.为克服这一缺点,本节介绍一种能灵活增加节点的牛顿插值公式. 因为在构造牛顿插值公式时,要用到差商,下面先介绍差商的概念.

一、差商的概念

定义 2.2 记 $f[x_i]=f(x_i)$ 称为函数 $f(x)$ 在 x_i 点的零阶差商.

$$f[x_i,x_j]=\frac{f[x_i]-f[x_j]}{x_i-x_j}$$

称为函数 $f(x)$ 在 x_i,x_j 点的一阶差商.一阶差商的差商

$$f[x_i,x_j,x_k]=\frac{f[x_i,x_j]-f[x_j,x_k]}{x_i-x_k}$$

称为 $f(x)$ 在 x_i,x_j,x_k 点的二阶差商.一般地,$k-1$ 阶差商的差商

$$f[x_0,x_1,\cdots,x_{k-1},x_k]=\frac{f[x_0,x_1,\cdots,x_{k-2},x_k]-f[x_0,x_1,\cdots,x_{k-1}]}{x_k-x_{k-1}}$$

称为 $f(x)$ 在点 $x_0,x_1,\cdots,x_{k-1},x_k$ 的 k 阶差商.

差商有如下一些基本性质：

（1）差商关于所含的节点是对称的.

例如，在 k 阶差商 $f[x_0, x_1, \cdots, x_k]$ 中，任意调换节点 x_i 与 x_j 的次序，其值不变，即

$$f[x_0, \cdots, x_i, \cdots, x_j, \cdots, x_k] = f[x_0, \cdots, x_j, \cdots, x_i, \cdots, x_k]$$

（2）差商可以表示为函数值的线性组合.

例如，函数 $f(x)$ 的 k 阶差商 $f[x_0, x_1, \cdots, x_k]$ 可由函数值 $f(x_0), f(x_1), \cdots, f(x_k)$ 的线性组合表示，即

$$f[x_0, x_1, \cdots, x_k] = \sum_{j=0}^{k} \frac{f(x_j)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_k)}$$

（3）如果函数 $f(x)$ 的 k 阶差商 $f[x_0, x_1, \cdots, x_{k-1}, x]$ 是 x 的 m 次多项式，则其 $k+1$ 阶差商 $f[x_0, x_1, \cdots, x_{k-1}, x_k, x]$ 是 x 的 $m-1$ 次多项式.

事实上，由差商的定义

$$f[x_0, x_1, \cdots, x_{k-1}, x_k, x]$$

$$= \frac{f[x_0, x_1, \cdots, x_{k-1}, x] - f[x_0, x_1, \cdots, x_{k-1}, x_k]}{x - x_k}$$

上式右端分子仍是 x 的 m 次多项式，且在 $x = x_k$ 时，其值为零，即分子中含有因子 $(x - x_k)$，所以分子、分母可约去公因子 $(x - x_k)$，故上式右端必是 $m-1$ 次多项式.

由此可推出：$f(x)$ 是 n 次多项式时，其 k 阶差商 $f[x_0, x_1, \cdots, x_{k-1}, x]$ 在 $k \leqslant n$ 时是 $n-k$ 次多项式；在 $k > n$ 时恒为零.

（4）设 $f(x)$ 在含有 $x_0, x_1, \cdots, x_n$ 的区间 $[a, b]$ 上具有 n 阶导数，则在这一区间内至少有一点 ξ，使

$$f[x_0, x_1, \cdots, x_n] = \frac{f^{(n)}(\xi)}{n!}, \quad \xi \in (a, b) \qquad (2.13)$$

上式揭示了差商与导数之间的关系.

二、牛顿插值公式

由各阶差商的定义可以得到如下的一组式子:

$$\begin{cases} f[x]=f[x_0]+(x-x_0)f[x,x_0] \\ f[x,x_0]=f[x_0,x_1]+(x-x_1)f[x,x_0,x_1] \\ f[x,x_0,x_1]=f[x_0,x_1,x_2]+(x-x_2)f[x,x_0,x_1,x_2] \\ \quad\cdots \\ f[x,x_0,x_1,\cdots,x_{n-1}]=f[x_0,x_1,\cdots,x_n]+(x-x_n)f[x,x_0,\cdots,x_n] \end{cases}$$

依次由下而上将后一式代入前一式,最后可得

$$f(x)=N_n(x)+R_n(x)$$

其中

$$\begin{aligned} N_n(x)=&f(x_0)+(x-x_0)f[x_0,x_1] \\ &+(x-x_0)(x-x_1)f[x_0,x_1,x_2]+\cdots \\ &+(x-x_0)(x-x_1)\cdots(x-x_{n-1})f[x_0,x_1,\cdots,x_n] \end{aligned} \qquad (2.14)$$

$$\begin{aligned} R_n(x)=&f(x)-N_n(x) \\ =&(x-x_0)(x-x_1)\cdots(x-x_n)f[x,x_0,x_1,\cdots,x_n] \end{aligned} \qquad (2.15)$$

由式(2.14)确定的多项式 $N_n(x)$ 显然满足插值条件:

$$N_n(x_i)=f(x_i),(i=0,1,2,\cdots,n)$$

且次数不超过 n 次. 根据插值多项式的唯一性,可知 $N_n(x)$ 恒等于拉格朗日插值多项式 $L_n(x)$,即 $N_n(x)\equiv L_n(x)$,只是形式上不同. $N_n(x)$ 称为 n 次牛顿插值多项式.

当函数 $f(x)$ 具有 $n+1$ 阶导数时,比较余项式(2.10)与式(2.15)可推出差商与导数的关系式(2.13). 但式(2.15)更具有一般性,它在 $f(x)$ 是由离散数据确定或导数不存在时仍有意义.

牛顿插值公式 $N_n(x)$ 相对拉格朗日插值公式的一个明显的

优点是，每增加一个插值节点，只要在原牛顿插值公式中增添一项便可形成高一次的插值公式. 一般地，有递推公式：

$$N_{k+1}(x)=N_k(x)+(x-x_0)(x-x_1)\cdots(x-x_k)f[x_0,x_1,\cdots,x_{k+1}] \tag{2.16}$$

式中，$N_k(x)$为 $f(x)$在节点 $x_0,x_1,\cdots,x_k$ 上的 k 次牛顿插值公式，$N_{k+1}(x)$为 $f(x)$在节点 $x_0,x_1,\cdots,x_k,x_{k+1}$ 上的 $k+1$ 次牛顿插值公式. 递推公式(2.16)指出，要计算增加一个节点 x_{k+1} 的牛顿插值公式 $N_{k+1}(x)$，只要在 $N_k(x)$中增添一项就可以了. 所以牛顿插值公式具有灵活增加节点的优点，在实际计算时会感到很方便.

从公式(2.14)中还可看到，牛顿插值公式中各项系数就是函数 $f(x)$的各阶差商：

$$f[x_0],f[x_0,x_1],\cdots,f[x_0,x_1,\cdots,x_n]$$

因此，在构造牛顿插值公式时，常常先把差商列成如下形状的表格，称为差商表(表 2-1).

表 2-1　　差 商 表

k	x_k	$f(x_k)$	一阶差商	二阶差商	三阶差商	……
0	x_0	$f(x_0)$				
			$f[x_0,x_1]$			
1	x_1	$f(x_1)$		$f[x_0,x_1,x_2]$		
			$f[x_1,x_2]$		$f[x_0,x_1,x_2,x_3]$	
2	x_2	$f(x_2)$		$f[x_1,x_2,x_3]$		…
			$f[x_2,x_3]$		$f[x_1,x_2,x_3,x_4]$	
3	x_3	$f(x_3)$		$f[x_2,x_3,x_4]$		
			$f[x_3,x_4]$		⋮	
4	x_4	$f(x_4)$		⋮		
			⋮			
⋮	⋮	⋮				

显然，$N_n(x)$的表达式(2.14)中各项的系数就是差商表中斜线上所对应的各阶差商，很有规律，按箭头所示，就容易写出各阶牛顿插值多项式. 而且，由差商的性质知，如果一个列表函数在差商表中的 k 阶差商接近于常数，那么用 k 次多项式去近似它是合理的.

例 2.2 用牛顿插值公式计算例 2.1 中的 $\ln 11.75$.

解 如果仍取节点 $x_0=11, x_1=12, x_2=13$，作二次牛顿插值多项式，则由给定的数据，先作差商表如下：

k	x_k	$f(x_k)=\ln x_k$	一阶差商	二阶差商
0	11	2.3979		
			0.0870	
1	12	2.4849		−0.0035
			0.0800	
2	13	2.5649		

以表中斜线上对应的数字为系数，按照公式(2.14)写出所求的二次牛顿插值多项式为

$$N_2(x)=2.3979+0.0870(x-11)-0.0035(x-11)(x-12)$$

所以

$$\begin{aligned}\ln 11.75 \approx N_2(11.75) &= 2.3979+0.0870(11.75-11)\\ &\quad -0.0035(11.75-11)(11.75-12)\\ &\approx 2.4638.\end{aligned}$$

三、差分与等距节点的插值公式

牛顿插值公式一般常应用于非等距节点的情形，但实际应用时，为了方便常采用等距节点，这时若利用差分可以进一步简化牛顿插值公式，导出计算上更为有效的等距节点的插值公式.

设已知函数 $f(x)$在等距节点 $x_i=x_0+ih,(i=0,1,2,\cdots,n)$上的函数值为 $f(x_i)=f_i$，式中 $h(>0)$称为步长.

定义 2.3 称 $\Delta f_i=f(x_i+h)-f(x_i)=f_{i+1}-f_i$ 为函数

$f(x)$在点 x_i 处步长为 h 的一阶向前差分.

称 $\Delta^2 f_i=\Delta(\Delta f_i)=\Delta f_{i+1}-\Delta f_i$ 为函数 $f(x)$在点 x_i 处的二阶向前差分. 一般地,设 $n-1$ 阶差分已定义,则称

$$\Delta^n f_i=\Delta(\Delta^{n-1} f_i)=\Delta^{n-1} f_{i+1}-\Delta^{n-1} f_i,\ (n=2,3,\cdots)$$

为函数 $f(x)$在点 x_i 处的 n 阶向前差分. 并规定

$$\Delta^0 f_i=f_i$$

为 $f(x)$在点 x_i 处的零阶差分.

类似地,还可以定义向后差分和中心差分.

由差分的定义,也可列出类同于差商表的差分表. 并可导出差分与差商的关系. 例如,k 阶差商与 k 阶向前差分之间有关系

$$f[x_0,x_1,\cdots,x_k]=\frac{\Delta^k f_0}{k!\,h^k},\quad (k=1,2,\cdots,n) \tag{2.17}$$

利用差分与差商的关系式(2.17),在牛顿插值公式(2.14)中将差商替换为差分,并令 $x=x_0+th$,$(0\leqslant t\leqslant 1)$,可得

$$N_n(x)=N_n(x_0+th)=f_0+\frac{t}{1!}\Delta f_0+\frac{t(t-1)}{2!}\Delta^2 f_0$$

$$+\cdots+\frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n f_0 \tag{2.18}$$

这个用各阶向前差分表示的插值公式(2.18)就称为牛顿向前插值公式. 其余项可由拉格朗日余项推导得

$$R_n(x)=R_n(x_0+th)=\frac{t(t-1)(t-2)\cdots(t-n)}{(n+1)!}h^{n+1}f^{(n+1)}(\xi) \tag{2.19}$$

2.4 埃尔米特(Hermite)插值

前面介绍的代数插值,只要求插值多项式 $P_n(x)$满足插值条

件(1.1),但是这种插值多项式往往还不能全面反映被插值函数 $f(x)$的性态,许多实际问题不但要求插值函数与被插值函数在各节点处的函数值相同,而且还要求插值函数在某些节点或全部节点上与 $f(x)$的导数值也相等,甚至要求高阶导数值也相等,这样的插值函数一定能更好地逼近函数 $f(x)$,我们称满足这种要求的插值问题为埃尔米特插值问题.

定义 2.4 设已知函数 $f(x)$在 $n+1$ 个互异的点 x_i $(i=0,1,2,\cdots,n)$处的函数值 $f(x_i)$和直到 p 阶的导数值 $f^{(p)}(x_i)$ $(p=1,2,\cdots)$ 若存在函数 $H(x)$满足条件:

$$H(x_i)=f(x_i),\ H^{(p)}(x_i)=f^{(p)}(x_i),\quad \begin{pmatrix} i=0,1,2,\cdots,n \\ p=1,2,\cdots \end{pmatrix} \tag{2.20}$$

则称 $H(x)$为 $f(x)$在 $n+1$ 个节点 x_i 上的埃尔米特插值函数.

若 $H(x)$是代数多项式,则称 $H(x)$为埃尔米特插值多项式.

在本节中,我们不去讨论上述一般的埃尔米特插值问题,而只是讨论下面这种最简单,但也是最常用的情形.

设已知函数 $f(x)$在插值区间$[a,b]$上 $n+1$ 个互异的节点 $x_0,x_1,\cdots,x_n$ 上的函数值 $f(x_i)=f_i$ 及一阶导数值 $f'(x_i)=f'_i$,$(i=0,1,2,\cdots,n)$,要求构造一个插值函数 $H(x)$,使得满足条件:

(1) $H(x)$是一个次数不超过 $2n+1$ 次的多项式;

(2) $H(x_i)=f(x_i),\quad H'(x_i)=f'(x_i),\quad (i=0,1,\cdots,n).$ (2.21)

并简称为埃尔米特插值问题,满足条件式(2.21)的多项式 $H(x)$称为埃尔米特插值多项式.

这个问题的几何意义是十分明显的,即要求代数曲线 $y=H(x)$与函数曲线 $y=f(x)$不仅在 $n+1$ 个互异的点(x_i,y_i)处完全重合,而且还有公切线,这时 $x_0,x_1,\cdots,x_n$ 称为二重节点.

定理 2.4 满足条件式(2.21)的埃尔米特插值问题的解 $H(x)$是存在且唯一的.

证明 Ⅰ.存在性:我们从构造拉格朗日插值多项式得到启发,拉格朗日插值多项式 $L_n(x)$是由 $n+1$ 个拉格朗日基函数 $l_k(x)$线性组合构成的:

$$L_n(x)=\sum_{k=0}^{n} l_k(x) f_k$$

现在如果也能构造两组次数都是 $2n+1$ 次的特殊多项式,记为 $h_k(x)$和 $\bar{h}_k(x)$,$(k=0,1,\cdots,n)$,使它们分别满足条件:

$$\begin{aligned} &h_k(x_i)=\delta_{ki},\ h_k'(x_i)=0 \\ &\bar{h}_k(x_i)=0\quad,\ \bar{h}_k'(x_i)=\delta_{ki},\quad (i,k=0,1,2,\cdots,n) \end{aligned} \tag{2.22}$$

这里 $h_k(x)$,$\bar{h}_k(x)$也称为插值基函数,共有 $2n+2$ 个,δ_{ki}是 Kronecker 记号,其意义同前.显然

$$H(x)=\sum_{k=0}^{n} h_k(x) f_k+\sum_{k=0}^{n} \bar{h}_k(x) f_k' \tag{2.23}$$

就是满足插值问题条件式(2.21)的埃尔米特插值问题的解.因此,下面要解决的问题是如何构造满足条件式(2.22)的 $2n+2$ 个插值基函数 $h_k(x)$和 $\bar{h}_k(x)$,$(k=0,1,2,\cdots,n)$.

(1) 构造 $h_k(x)$,$(k=0,1,2,\cdots,n)$.

由条件 $h_k(x_i)=h_k'(x_i)=0$,$(i\neq k)$知,x_i(当 $i\neq k$)都是$h_k(x)$的二重零点,而已知拉格朗日插值基函数 $l_k(x)$是 n 次多项式,且满足 $l_k(x_i)=\delta_{ki}$,因此可设

$$h_k(x)=(ax+b)l_k^2(x),\quad (k=0,1,\cdots,n)$$

其中,a,b 为待定常数.显然,当 $i\neq k$ 时,$h_k(x)$已满足条件式(2.22)中的 $h_k(x_i)=h_k'(x_i)=0$.现在只要适当选取常数 a,b,使当 $i=k$ 时,$h_k(x)$也满足条件式(2.22),即满足

$$h_k(x_k)=1, h_k'(x_k)=0$$

即可得到确定 a,b 的两个方程：

$$\begin{cases} ax_k+b=1 \\ a+2l_k'(x_k)=0 \end{cases} \quad \text{解出} \quad \begin{cases} a=-2l_k'(x_k) \\ b=1+2l_k'(x_k)x_k \end{cases}$$

于是得 $$h_k(x)=[(-2l_k'(x_k)x+1+2l_k'(x_k)x_k]l_k^2(x)$$
$$=[1-2l_k'(x_k)(x-x_k)]l_k^2(x)$$

(2) 构造 $\bar{h}_k(x)$，$(k=0,1,\cdots,n)$.

类似于上述 $h_k(x)$ 的构造，由条件 $\bar{h}_k(x_i)=\bar{h}_k'(x_i)=0$ $(i\neq k)$，可设 $\bar{h}_k(x)=A(x-x_k)l_k^2(x)$，其中，$A$ 为待定常数. 显然 $\bar{h}_k(x)$是 $2n+1$ 次多项式，且当 $i\neq k$ 时已满足条件式(2.22)，现由条件 $\bar{h}_k(x_k)=1$ 可直接确定 $A=1$，所以 $\bar{h}_k(x)=(x-x_k)l_k^2(x)$.

到此为止，已构造出 $2n+2$ 个满足插值条件式(2.22)的插值基函数 $h_k(x)$和 $\bar{h}_k(x)$，则将其代入 $H(x)$的表达式(2.23)，即得满足插值条件(1)，(2)的埃尔米特插值问题的解：

$$H(x)=\sum_{k=0}^{n}[1-2l_k'(x_k)(x-x_k)]l_k^2(x)f_k+\sum_{k=0}^{n}(x-x_k)l_k^2(x)f_k' \tag{2.24}$$

以上用构造性方法证明了满足条件式(2.21)为埃尔米特插值问题的解$H(x)$的存在性，且具体地给出了满足条件式(2.21)的插值多项式$H(x)$的求法.

Ⅱ. 唯一性：假设另有一个 $2n+1$ 次多项式 $\overline{H}(x)$也满足插值条件式(2.21)，则有

$$H(x_i)=\overline{H}(x_i)=f_i;\ H'(x_i)=\overline{H}'(x_i)=f_i',\ (i=0,1,\cdots,n)$$

于是函数

$$\varphi(x)=H(x)-\overline{H}(x)$$

也是次数不超过$(2n+1)$次的多项式，且有

$$\varphi(x_i)=H(x_i)-\overline{H}(x_i)=f_i-f_i=0$$

$$\varphi'(x_i)=H'(x_i)-\overline{H}'(x_i)=f_i'-f_i'=0,\ (i=0,1,\cdots,n)$$

即 $\varphi(x)$ 至少有 $2n+2$ 个根，而 $\varphi(x)$ 是次数不超过 $2n+1$ 次的多项式，所以只能 $\varphi(x)\equiv 0$，即 $H(x)\equiv\overline{H}(x)$. 唯一性得证.

用 $H(x)\approx f(x)$，在非节点处一般有误差，下面给出其余项定理.

定理 2.5 若 $f(x)$ 在插值区间$[a,b]$上有 $2n+2$ 阶导数，则对于任何 $x\in[a,b]$，满足条件式(2.21)的埃尔米特插值问题的余项为

$$R(x)=f(x)-H(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}[\pi(x)]^2 \tag{2.25}$$

其中，$\pi(x)=\prod_{j=0}^{n}(x-x_j)$，$\xi\in(a,b)$，且 ξ 与 x 有关.

证明 类同于拉格朗日插值余项的证明.

当 $x=x_i$ 时，定理的结论显然成立. 今设 x 异于 x_i，构造辅助函数

$$g(t)=f(t)-H(t)-k(x)[\pi(t)]^2$$

则易知，$g(t)$ 在$[a,b]$上至少有 $n+1$ 个二重零点 $x_i(i=0,1,\cdots,n)$ 和一个单零点 x. 由罗尔定理知，$g'(t)$ 在点 $x,x_0,x_1,\cdots,x_n$ 之间至少有 $n+1$ 个零点，且 $x_i(i=0,1,\cdots,n)$ 仍然是 $g'(t)$ 的零点，故 $g'(t)$ 在区间(a,b)内至少有 $2n+2$ 个零点. 继续反复使用罗尔定理，可知 $g^{(2n+2)}(t)$ 在(a,b)内至少有一个零点 ξ，使

$$g^{(2n+2)}(\xi)=0$$

而
$$g^{(2n+2)}(\xi)=f^{(2n+2)}(\xi)-k(x)(2n+2)!=0$$

于是得
$$k(x)=f^{(2n+2)}(\xi)/(2n+2)!$$

也就证得了余项公式(2.25).

实际使用较多的是三次埃尔米特插值多项式,即在式(2.24)中令 $n=1$ 得

$$H_3(x)=\left(1+2\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_1}{x_0-x_1}\right)^2 f_0+\left(1+2\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_0}{x_1-x_0}\right)^2 f_1$$

$$+(x-x_0)\left(\frac{x-x_1}{x_0-x_1}\right)^2 f_0'+(x-x_1)\left(\frac{x-x_0}{x_1-x_0}\right)^2 f_1' \quad (2.26)$$

其余项为

$$R_3(x)=f(x)-H_3(x)=\frac{1}{4!}f^{(4)}(\xi)(x-x_0)^2(x-x_1)^2$$

$$\xi\in(x_0,x_1) \qquad (2.27)$$

还需指出,埃尔米特插值问题解的形式可以有多种,插值条件也可由多种形式给出.例如已知$(n+1)$个互异节点上的函数值和某几点上的导数值,只要给出的插值条件数比要求的多项式次数多一个.实际求解时,常应用所谓“待定系数法”,得到的埃米尔特插值多项式也常有两种不同表达形式,分别称为①拉格朗日-埃尔米特插值多项式;② 牛顿-埃尔米特插值多项式.
下面举例说明.

例 4.1 若 $f(x)$在$[a,b]$上有三阶连续导数,且已知 $f(x)$在$[a,b]$上两个互异的节点 x_0,x_1 上的函数值 $f(x_0),f(x_1)$和一阶导数值 $f'(x_0)$.试求满足条件

$$H(x_0)=f(x_0),H(x_1)=f(x_1),H'(x_0)=f'(x_0)$$

的插值多项式 $H(x)$,并估计误差.

解 由给定的 3 个插值条件,显然可确定一个次数不超过 2 次的埃米尔特插值多项式 $H(x)$,又由 $H(x)$应满足插值条件:$H(x_i)=f(x_i),(i=0,1)$,而节点 x_0,x_1 上的一次牛顿插值函数 $N_1(x)$也满足插值条件$N_1(x_i)=f(x_i),(i=0,1)$,故可设

$$H(x)-N_1(x)=A(x-x_0)(x-x_1)$$

其中,A 为待定常数. 上式又可记为

$$\begin{aligned}H(x)&=N_1(x)+A(x-x_0)(x-x_1)\\&=f(x_0)+(x-x_0)f[x_0,x_1]+A(x-x_0)(x-x_1)\end{aligned}\tag{2.28}$$

为了确定常数 A,对上式求导,得

$$H'(x)=N_1'(x)+A[(x-x_1)+(x-x_0)]$$

令 $x=x_0$ 代入,且注意插值条件 $H'(x_0)=f'(x_0)$得

$$\begin{aligned}H'(x_0)&=N_1'(x_0)+A(x-x_1)=f[x_0,x_1]+A(x_0-x_1)\\&=f'(x_0)\end{aligned}$$

于是有

$$A=\frac{f[x_0,x_1]-f'(x_0)}{x_1-x_0}$$

将 A 代入式(2.28),即得所求的插值多项式为

$$\begin{aligned}H(x)=&f(x_0)+(x-x_0)f[x_0,x_1]\\&+\frac{f[x_0,x_1]-f'(x_0)}{x_1-x_0}(x-x_0)(x-x_1)\end{aligned}$$

因为上式是用牛顿插值多项式构造的,所以也称为牛顿-埃尔米特插值多项式. 当然也可先采用拉格朗日多项式构造 $L_1(x)$,再令

$$H(x)=L_1(x)+A(x-x_0)(x-x_1)$$

同样可得到满足相同插值条件的 $H(x)$的另一种形式

$$H(x)=\left(\frac{x-x_0+x_1-x_0}{x_1-x_0}\right)\left(\frac{x_1-x}{x_1-x_0}\right)f(x_0)$$

$$+\left(\frac{x-x_0}{x_1-x_0}\right)^2 f(x_1)+\left(\frac{x_1-x}{x_1-x_0}\right)(x-x_0)f'(x_0)$$

上式也称为拉格朗日-埃尔米特插值多项式. 类同于埃尔米特插值的余项定理的证明,可推得其误差为

$$R(x)=f(x)-H(x)=\frac{1}{3!}f'''(\xi)(x-x_0)^2(x-x_1),\xi\in(a,b)$$

2.5 三次样条插值

利用插值多项式近似函数时,人们容易产生一种错觉,认为插值多项式的次数越高精度越好,但实际并非如此. 事实上,当插值节点无限增多时,次数趋于无穷的插值多项式不一定收敛到被插值函数. 一个有名的实例是在 20 世纪初由德国数学家龙格(Runge)发现的:函数 $f(x)=\frac{1}{1+x^2}$ 在$[-5,5]$上取$(n+1)$个等距节点 $x_i=-5+10\times\frac{i}{n}$,($i=0,1,2,\cdots,n$),作 n 次插值多项式 $L_n(x)$,当 $n\to\infty$时,在$[-5,5]$上并不收敛于 $f(x)$. 特别在 $x=\pm 5$ 的附近,$L_n(x)$偏离 $f(x)$很远,这种现象称为龙格现象. 因此,在应用中一般不用高次($n>7$)插值多项式来逼近函数. 但在解决实际问题时,人们总希望将所得到的数据点用得越多越好,解决这一矛盾的办法是改用分段低次插值. 所谓分段低次插值,即用分段多项式来代替单个多项式作插值. 具体说,就是在实际进行插值时,先把整个插值区间分成若干个小区间,然后在每个小区间上分别用低次插值多项式进行插值. 例如可用线性插值或抛物线插值等,然后再将每个小区间上的插值函数拼接在一起,作为整个插值区间上的插值函数. 这样一种插值方法就叫做分段低次插值. 显然,分段低次插值具有公式简单、节省运算量、稳定性好、收敛性有保证等优点,只要每一个小区间的长度取得足够小,分段低次插值

总能满足所要求的精度. 但是分段低次插值的一个明显的缺点是不能保证整条插值曲线在连接点处的光滑性，即导数在节点处不一定连续，如分段线性插值，其图形是锯齿形的折线，它虽然连续，但一阶导数在节点 $x_i(i=1,2,\cdots,n-1)$ 处都不存在，因而极不光滑，这在使用上往往不能满足要求. 例如，在船体、飞机等外型曲线的设计中，不仅要求曲线连续，而且还要求曲线的曲率连续，这就要求分段插值函数在整个区间上具有连续的二阶导数 . 为此，下面介绍三次样条插值.

一、三次样条插值函数的概念

样条(Spline)这个词本来是指在飞机和船舶的制造过程中为了描绘出光滑的外形曲线所用的一种绘图工具. 它是一种富有弹性的细长条，使用时，用压铁固定在一些给定的点(称为样点)上，在其他地方任它自由弯曲，然后依样画下光滑的曲线，就称为样条曲线. 从数学上看，在小挠度的假定下，它实际上是由一段一段的三次多项式曲线拼接而成的，在拼接处，不仅函数自身是连续的，而且它的一阶导数和二阶导数也是连续的(可是三阶导数一般不连续). 所以样条曲线具有非常好的光滑性. 从数学上加以概括就得到三次样条插值函数这一概念.

近几十年来，函数逼近在理论研究和实际应用中均获得重大的进展，其中非常流行和十分活跃的分支就是样条函数，它已经以各种方式应用到数据拟合、数值微分、数值积分、微分方程和积分方程的数值求解中，且三次样条插值函数的计算也是比较简单的.

定义 2.5　对于给定的函数表

x	x_0	x_1	x_2	$\cdots$	x_n
$y=f(x)$	y_0	y_1	y_2	$\cdots$	y_n

其中

$$a=x_0<x_1<x_2<\cdots<x_n=b$$

若函数 $S(x)$ 满足条件：

(1) $S(x)$ 在每一个子区间 $[x_{j-1},x_j]$，$(j=1,2,\cdots,n)$ 上是一个三次多项式；

(2) $S(x)$ 在每一个内接点 x_j，$(j=1,2,\cdots,n-1)$ 上具有直到二阶的连续导数，即 $S(x)\in c^2[a,b]$，则称 $S(x)$ 为节点 $x_0,x_1,\cdots,x_n$ 上的三次样条函数.

若 $S(x)$ 在所有节点上还满足插值条件：

(3) $S(x_j)=y_j$，　$(j=0,1,2,\cdots,n)$，

则称 $S(x)$ 为三次样条插值函数.

由此可见，三次样条插值函数就是全部通过样点的二阶连续可微的分段三次多项式函数.

由定义 2.5 的条件(1)知，$S(x)$ 在每一个小区间 $[x_{j-1},x_j]$ 上是一个三次多项式，若记为 $S_j(x)$，则可设

$$S_j(x)=a_jx^3+b_jx^2+c_jx+d_j,\quad (j=1,2,\cdots,n)$$

从而有

$$S(x)=\begin{cases}S_1(x)\\S_2(x)\\\vdots\\S_n(x)\end{cases}$$

这是一个分段函数. 显然，$S_j(x)$ 由 a_j,b_j,c_j,d_j 唯一确定，因此要确定整个 $[a,b]$ 上的三次样条插值函数 $S(x)$，必须确定 $4n$ 个未知系数 $\{a_j,b_j,c_j,d_j\}$ $(j=1,2,\cdots,n)$，现在我们来分析一下，这 $4n$ 个未知系数能否由已知条件来确定.

首先，由定义 2.5 的条件(2)表明 $S(x)$，$S'(x)$，$S''(x)$ 在内节点 $x_1,x_2,\cdots,x_{n-1}$ 上是连续的，于是有

$$S(x_j-0)=S(x_j+0),\quad S'(x_j-0)=S'(x_j+0),$$

$$S''(x_j-0)=S''(x_j+0),\quad (j=1,2,\cdots,n-1)$$

由于这样的等式共有 $3(n-1)$ 个，从而提供了包含未知系数 a_j，b_j，c_j，d_j 的 $3n-3$ 个方程，由定义 2.5 的条件(3)

$$S(x_j)=y_j,\ (j=0,1,2,\cdots,n)$$

又提供了 $n+1$ 个方程，这样一共可以得到 $4n-2$ 个方程，但与要确定的 $4n$ 个未知系数相比较，还差两个条件，所以，对于给定的插值函数表，按定义 2.5 所得到的样条插值函数 $S(x)$ 中含有两个自由度，是个函数族. 要想从这个函数族中唯一地确定一个函数，还要补充两个条件. 通常是在区间 $[a,b]$ 的端点 $a=x_0$，$b=x_n$ 上各附加一个条件. 在端点上的条件称为边界条件. 常见的边界条件有如下三种：

(1) 第一种边界条件：给定端点处的一阶导数值，即

$$S'(x_0)=y'(x_0),\quad S'(x_n)=y'(x_n) \tag{2.29}$$

(2) 第二种边界条件：给定端点处的二阶导数值，即

$$S''(x_0)=y''(x_0),\quad S''(x_n)=y''(x_n) \tag{2.30}$$

作为特例，若 $S''(x_0)=S''(x_n)=0$，则称为自然边界，满足自然边界条件的样条函数称为自然样条. 它是通过所有数据点的插值函数中，总曲率最小的唯一函数，因此自然三次样条是插值所有数据点的最光滑的函数.

(3) 第三种边界条件：当被插值函数是以 $b-a$ 为周期的周期函数时，则要求 $S(x)$ 也是周期函数，这时边界条件可以写成

$$S(x_0+0)=S(x_n-0),$$

$$S'(x_0+0)=S'(x_n-0),$$

$$S''(x_0+0)=S''(x_n-0).$$

其实，真正起作用的是后两个等式.

补充了两个端点条件，即增加了两个方程，也即得到了关于

$4n$ 个未知系数$\{a_j, b_j, c_j, d_j\}$ $(j=1,2,\cdots,n)$的 $4n$ 个方程,可以证明,在上述三种边界条件下,插值问题的解 $S(x)$均存在且唯一. 从而有

定理 2.6 三次样条插值问题的解存在且唯一.

理论上,只要求解上述的 $4n$ 阶线代数方程组就可求得 $S(x)$,但在应用上,因方程组常常是病态的,且计算工作量太大,所以上述这种将样条插值函数的构造归结为求解线代数方程组的方法常常不用. 下面介绍一种计算工作量小得多的切实可行的构造$S(x)$的有效方法.

二、三次样条插值函数的求法

求三次样条插值函数 $S(x)$常用的有三弯矩法和三转角法,下面重点介绍构造三次样条插值函数的三弯矩法.

设对区间$[a,b]$的分划

$$a=x_0<x_1<\cdots<x_n=b \qquad \text{是任意的.}$$

记 $S''(x_j)=M_j$, $(j=0,1,2,\cdots,n)$, $h_j=x_j-x_{j-1}$

由于 $S(x)$在每个子区间$[x_{j-1},x_j]$上是一个三次多项式 $S_j(x)$,故 $S_j''(x)$在区间$[x_{j-1},x_j]$上是 x 的线性函数,且有 $S_j''(x_{j-1})=M_{j-1}$, $S_j''(x_j)=M_j$,用线性插值,可知其表达式为

$$S_j''(x)=\frac{x_j-x}{h_j}M_{j-1}+\frac{x-x_{j-1}}{h_j}M_j \tag{2.31}$$

将式(2.31)积分两次得

$$S_j(x)=\frac{(x_j-x)^3}{6h_j}M_{j-1}+\frac{(x-x_{j-1})^3}{6h_j}M_j+C_1x+C_2 \tag{2.32}$$

利用插值条件:$S_j(x_{j-1})=y_{j-1}$,$S_j(x_j)=y_j$确定积分常数 C_1 和 C_2,然后代入式(2.32)中并整理,得到 $S(x)$在$[x_{j-1}, x_j]$上的表达式为

$$S_j(x)=\frac{(x_j-x)^3}{6h_j}M_{j-1}+\frac{(x-x_{j-1})^3}{6h_j}M_j$$

$$+\left(y_{j-1}-\frac{M_{j-1}h_j^2}{6}\right)\frac{x_j-x}{h_j}+\left(y_j-\frac{M_jh_j^2}{6}\right)\frac{x-x_{j-1}}{h_j}$$

$$(j=1,2,\cdots,n) \tag{2.33}$$

由此可知,只要确定 $M_0,M_1,\cdots,M_n$ **这** $n+1$ **个待定参数,所求的三次样条插值函数** $S(x)$ **在各个子区间上的表达式就由式**(2.33)**确定,从而就求得了在整个**$[a,b]$**上的三次样条插值函数** $S(x)$**. 并由以上构造过程知,这时的** $S(x)$ **保证了逐段三次插值,即** $S(x)$ **的连续性,并保证了** $S''(x)$ **在内节点的连续性.**

为了确定 $M_0,M_1,\cdots,M_n$**,可利用函数** $S(x)$ **在插值区间**$[a,b]$**上各内接点** $x_j(j=1,2,\cdots,n-1)$ **处有一阶导数连续的条件:**

$$S'(x_j-0)=S'(x_j+0)$$

对 $S(x)$ **求导,由式**(2.33)**得在区间**$[x_{j-1},x_j]$**上有**

$$S_j'(x)=-\frac{(x_j-x)^2}{2h_j}M_{j-1}+\frac{(x-x_{j-1})^2}{2h_j}M_j+\frac{y_j-y_{j-1}}{h_j}-\frac{M_j-M_{j-1}}{6}h_j$$

从而在右端点 x_j **上有**

$$S_j'(x_j-0)=\frac{h_j}{2}M_j-\frac{h_j}{6}(M_j-M_{j-1})+\frac{y_j-y_{j-1}}{h_j}$$

$$=\frac{h_j}{6}M_{j-1}+\frac{h_j}{3}M_j+\frac{y_j-y_{j-1}}{h_j} \tag{2.34}$$

在左端点 x_{j-1} **上有**

$$S_j'(x_{j-1}+0)=-\frac{h_j}{2}M_{j-1}-\frac{h_j}{6}(M_j-M_{j-1})+\frac{y_j-y_{j-1}}{h_j}$$

$$=-\frac{h_j}{3}M_{j-1}-\frac{h_j}{6}M_j+\frac{y_j-y_{j-1}}{h_j} \tag{2.35}$$

则将式(2.35)中的 $j-1$ 改为 j 即得

$$S'_{j+1}(x_j+0)=-\frac{h_{j+1}}{3}M_j-\frac{h_{j+1}}{6}M_{j+1}+\frac{y_{j+1}-y_j}{h_{j+1}} \tag{2.36}$$

利用 $S'(x)$在内接点连续的条件：

$$S'_j(x_j-0)=S'_{j+1}(x_j+0)$$

就得到在 $n-1$ 个内接点 $x_j(j=1,2,\cdots,n-1)$处关于 $n+1$ 个参数 $M_j(j=0,1,2,\cdots,n)$的 $n-1$ 个方程式：

$$\mu_j M_{j-1}+2M_j+\lambda_j M_{j+1}=d_j,\quad (j=1,2,\cdots,n-1) \tag{2.37}$$

其中

$$\begin{cases}\lambda_j=\dfrac{h_{j+1}}{h_j+h_{j+1}}\\ \mu_j=1-\lambda_j=\dfrac{h_j}{h_j+h_{j+1}}\\ d_j=\dfrac{6}{h_j+h_{j+1}}\left(\dfrac{y_{j+1}-y_j}{h_{j+1}}-\dfrac{y_j-y_{j-1}}{h_j}\right)=6f[x_{j-1},x_j,x_{j+1}]\end{cases} \tag{2.38}$$

由于力学上将 M_j 解释为梁在截面 x_j 处的弯矩，故式(2.37)常称为三弯矩方程. 因式(2.37)是关于 $n+1$ 个参数 M_j 的 $n-1$ 个方程，所以有无穷多组解. 要唯一确定 $n+1$ 个未知数 M_j，尚须附加边界条件，即补充两个方程.

例如，(1)第一种边界条件：已知插值区间两端的一阶导数值：

$$S'(x_0)=y'_0,\ S'(x_n)=y'_n$$

利用式(2.34)和式(2.36)可得到以下两个方程：

$$2M_0+M_1=\frac{6}{h_1}\left(\frac{y_1-y_0}{h_1}-y'_0\right)\overset{\text{记}}{=\!=}d_0 \tag{2.39}$$

$$M_{n-1}+2M_n=\frac{6}{h_n}\left(y_n'-\frac{y_n-y_{n-1}}{h_n}\right)\stackrel{\text{记}}{=\!=}d_n \qquad (2.40)$$

将方程式(2.39),式(2.37),式(2.40)合在一起,构成了确定$n+1$个参数$M_0, M_1,\cdots,M_n$的$n+1$阶线代数方程组

$$\begin{bmatrix} 2 & 1 & & & & \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & & 1 & 2 \end{bmatrix}\begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix}=\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \qquad (2.41)$$

它的系数矩阵是严格对角占优的,故可证明有唯一解,并可用后面介绍的追赶法求解.

(2) 第二种边界条件:

$$S''(x_0)=y''(x_0),S''(x_n)=y''(x_n)$$

即已知$M_0=y_0''$, $M_n=y_n''$,所以方程组(2.41)只含有$n-1$个未知元$M_1,M_2,\cdots,M_{n-1}$,可以表示成

$$\begin{bmatrix} 2 & \lambda_1 & & & \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-2} & 2 & \lambda_{n-2} \\ & & & \mu_{n-1} & 2 \end{bmatrix}\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix}=\begin{bmatrix} d_1-\mu_1 y_0'' \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1}-\lambda_{n-1} y_n'' \end{bmatrix} \qquad (2.42)$$

特别当$M_0=M_n=0$,为自然边界条件,方程组(2.42)的右端成为$[d_1,d_2,\cdots,d_{n-1}]^{\mathrm{T}}$,其形式特别简单.

(3) 第三种边界条件:

$$S''(x_0+0)=S''(x_n-0),\quad S'(x_0+0)=S'(x_n-0)$$

可以导出两个方程

$$M_0=M_n,\quad \lambda_n M_1+\mu_n M_{n-1}+2M_n=d_n$$

其中，$\lambda_n=\dfrac{h_1}{h_1+h_n}$，$\mu_n=1-\lambda_n=\dfrac{h_n}{h_1+h_n}$，

$$d_n=\frac{6}{h_1+h_n}\left(\frac{y_1-y_0}{h_1}-\frac{y_n-y_{n-1}}{h_n}\right)$$

与式(2.37)合在一起构成确定 $M_1,M_2,\cdots,M_n$ 为未知元的线代数方程组

$$\begin{bmatrix} 2 & \lambda_1 & & \cdots & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \vdots \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ \lambda_n & & & \mu_n & 2 \end{bmatrix}\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix}=\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{2.43}$$

方程组(2.43)的系数矩阵也是非奇异的，因而此方程组也有唯一解.

综上所述，对于给定的函数表$(x_i,f(x_i))(i=0,1,2,\cdots,n)$满足第一(或第二，第三)种边界条件的三次样条插值函数 $S(x)$是存在且唯一的，且其计算过程可归纳如下：

(1) 根据给定的数据$(x_i,\ y_i)(i=0,1,2,\cdots,n)$及相应的边界条件建立方程组(2.41)或(2.42)或(2.43)；

(2) 解上述线性方程组，求出 $M_0,\ M_1,\cdots,M_n$；

(3) 把求出的 $M_0,\ M_1,\ \cdots,M_n$ 代入 $S_j(x)$的表达式(2.33)，即得 $S(x)$在每一个小区间$[x_{j-1},x_j]$上的分段表达式；

(4) 整个区间$[a,b]$上的三次样条插值函数可表示为

$$S(x)=\begin{cases} S_1(x) \\ S_2(x) \\ \vdots \\ S_n(x) \end{cases}.$$

例 2.3　已知函数表

x_i	1	2	4	5
$y_i=f(x_i)$	1	3	4	2

在区间[1,5]上用三弯矩法求满足上述函数表所给出的插值条件的三次自然样条插值函数.

解　因为这是在第二种边界条件下的插值问题,故确定 M_0, M_1, M_2, M_3 **的方程组形如式**(2.42),**由已知自然边界条件,有** $M_0=M_3=0$, **则得求解** M_1, M_2 **的方程组为**

$$\begin{bmatrix} 2 & \lambda_1 \\ \mu_2 & 2 \end{bmatrix}\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}=\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{2.44}$$

为确定方程组的右端项 d_1, d_2, **先作差商表:**

i	x_i	$f(x_i)$	一阶差商	二阶差商
0	1	1		
			2	
1	2	3		$-\frac{1}{2}$
			$\frac{1}{2}$	
2	4	4		$-\frac{5}{6}$
			-2	
3	5	2		

得
$$d_1=6f[x_0,x_1,x_2]=6\times\left(-\frac{1}{2}\right)=-3$$

$$d_2=6f[x_1,x_2,x_3]=6\times\left(-\frac{5}{6}\right)=-5$$

再由 $x_0=1$, $x_1=2$, $x_2=4$, $x_3=5$ **得** $h_1=x_1-x_0=1$, $h_2=2$, $h_3=1$. **所以** $\lambda_1=\frac{h_2}{h_1+h_2}=\frac{2}{3}$, $\mu_2=\frac{h_2}{h_2+h_3}=\frac{2}{3}$, **代入方程组**(2.44). **得**

$$\begin{cases} 2M_1+\frac{2}{3}M_2=-3 \\ \frac{2}{3}M_1+2M_2=-5 \end{cases} \quad \text{解得} \quad \begin{cases} M_1=-\frac{3}{4} \\ M_2=-\frac{9}{4} \end{cases}$$

代入式(2.33),经整理得所求的三次自然样条插值函数 $S(x)$ 在区间$[1,5]$上的表达式为

$$S(x)=\begin{cases} -\frac{1}{8}x^3+\frac{3}{8}x^2+\frac{7}{4}x-1, & (1\leqslant x\leqslant 2) \\ -\frac{1}{8}x^3+\frac{3}{8}x^2+\frac{7}{4}x-1, & (2\leqslant x\leqslant 4) \\ \frac{3}{8}x^3-\frac{45}{8}x^2+\frac{103}{4}x-33, & (4\leqslant x\leqslant 5) \end{cases}.$$

利用三次样条插值,不会出现拉格朗日插值时的那种"龙格现象".为了提高插值的精确度,样条插值可用增加节点的办法来实现.可以证明,对于三次样条插值函数,当插值节点逐渐加密时,不但样条函数收敛于函数本身,而且其导数也同样收敛到函数的导数,这种性质要比多项式插值优越得多.

2.6 B-样条函数

上面讨论的三次样条插值函数 $S(x)$ 是一个分段三次多项式函数,在每一个子区间$[x_{i-1},x_i]$上 $S(x)$ 都有一个独立的表达式,这在应用上和理论分析中有时不方便,而如果利用基样条表示,往往更为方便.为此,对于区间$[a,b]$的某种分划:

$$\Delta: a=x_0<x_1<\cdots<x_n=b$$

先给出 m 次样条函数的概念,再构造 m 次样条函数空间的一个有用的基函数——**B**-样条函数.

定义 2.6 设$[a,b]$上给定一个分划:

$$\Delta: a=x_0<x_1<\cdots<x_n=b$$

如果函数 $S(x)$ 满足下列条件：

(1) $S(x)$ 在区间 $[a,b]$ 上具有直到 $(m-1)$ 阶连续导数，记为

$$S(x)\in c^{m-1}[a,b];$$

(2) $S(x)$ 在每个子区间 $[x_{i-1}, x_i]$ $(i=1,2,\cdots,n)$ 上是 m 次代数多项式；

则称 $S(x)$ 是关于节点分划 Δ 的 m 次样条函数.

设满足上述条件的 m 次样条函数的全体组成的集合为 $S(m,\Delta)$，容易验证 $S(m,\Delta)$ 是一个线性空间，并且它的维数为 $n+m$ 维，这是因为：对于分划 Δ，在区间 $[x_{i-1}, x_i]$ 上，$S(x)$ 是一个 m 次多项式：

$$P_m^i(x)=a_0^i+a_1^i x+\cdots+a_m^i x^m, \ (i=1,2,\cdots,n)$$

其中上标 i 表示第 i 个多项式，显然，每个多项式由它的 $m+1$ 个系数 $a_0^i, a_1^i, \cdots, a_m^i$ 完全确定，即每个多项式有 $m+1$ 个参数，因此 m 次样条函数空间 $S(m,\Delta)$ 至多有 $(m+1)\times n$ 个自由参数. 又由 $S(x)$ 在内接点处满足连续性条件，则共有 $m\times(n-1)$ 个约束条件，所以 $S(m,\Delta)$ 的维数至多为 $(m+1)\times n-m\times(n-1)=n+m$.

定义截幂函数为 $\quad x_+^m=\begin{cases} x^m, & x\geqslant 0 \\ 0, & x<0 \end{cases}$

下面证明，$S(m,\Delta)$ 中的 $m+n$ 个样条函数

$$x^k, \quad k=0,1,2,\cdots,m$$

$$(x-x_j)_+^m, \ j=1, 2, \cdots, n-1 \tag{2.45}$$

在区间 $[a,b]$ 上线性无关，从而可得出 $S(m,\Delta)$ 的维数为 $m+n$，也即 $S(m,\Delta)$ 是 $m+n$ 维的线性空间.

用反证法. 假定式(2.45)中的 $m+n$ 个函数在 $[a,b]$ 上线性相

关，即存在不全为零的常数 $a_k(k=0,1,\cdots,m)$ 与 $c_j(j=1,2,\cdots,n-1)$，使

$$\sum_{k=0}^{m} a_k x^k + \sum_{j=1}^{n-1} c_j (x-x_j)_+^m = 0 \tag{2.46}$$

这样，对于 $x<x_1$，由截幂函数定义，等式(2.46)变成

$$a_0 + a_1 x + \cdots + a_m x^m = 0$$

由 $1, x, x^2, \cdots, x^m$ 的线性无关性，可推出 $a_k=0$，$(k=0,1,\cdots,m)$.

进而，对于 $x\in[x_1, x_2]$，得到 $c_1(x-x_1)^m=0$，从而有 $c_1=0$，依此类推，可得到所有 $c_j=0$，$(j=1,2,\cdots,n-1)$，这就证明了式(2.45)的 $m+n$ 个函数线性无关.

下面作出线性空间 $S(m,\Delta)$ 的一组特殊基函数，这组基函数就是 **B-** 样条函数.

为了构造 B- 样条函数，先对分划 Δ 加入新节点扩展为

$$x_{-m}<\cdots<x_{-1}<a=x_0<x_1<\cdots<x_n=b<x_{n+1}<\cdots<x_{n+m} \tag{2.47}$$

令
$$\varphi_m(t;x)=(t-x)_+^m$$

x 视为参数，$\varphi_m(t;x)$ 是 t 的函数，当

$$t=x_{-m},\cdots,x_{-1},x_0, x_1,\cdots,x_n, x_{n+1},\cdots,x_{n+m}$$

则 $\varphi_m(x_{-m};x)$，$\cdots$，$\varphi_m(x_{-1};x)$，$\varphi_m(x_0;x)$，$\cdots$，$\varphi_m(x_{n+m};x)$ 都是关于分划 Δ 的样条函数，记

$$\varphi_m(t)=\varphi_m(t;x)$$

关于 $t=x_j, x_{j+1}, \cdots, x_{j+m+1}$ 所作的 $m+1$ 阶差商为 $\varphi_m[x_j, x_{j+1}, \cdots, x_{j+m+1}]$.

定义 2.7 设 $\{x_i\}$ 是节点序列，令 $\varphi_m(t)=(t-x)_+$，函数 $(x_{j+m+1}-x_j)\varphi_m(t)$ 关于 $t=x_j,\cdots,x_{j+m+1}$ 的 $m+1$ 阶差商

$$B_{j,m}=(x_{j+m+1}-x_j)\varphi_m[x_j,x_{j+1},\cdots,x_{j+m+1}]$$

$$(j=-m,\ -m+1,\ \cdots,n-1) \tag{2.48}$$

称为第 j 个 m 次 *B*-样条函数，简称 *B*-样条函数. 在不发生误会的场合也常记为 $B_j(x)$.

利用差商的性质：

$$\varphi_m[x_j,x_{j+1},\ \cdots,x_{j+m+1}]=\sum_{k=j}^{j+m+1}\frac{(x_k-x)_+^m}{\omega'_{m,j}(x_k)}$$

其中 $$\omega_{m,j}(t)=(t-x_j)(t-x_{j+1})\cdots(t-x_{j+m+1})$$

得到 $$B_{j,m}(x)=(x_{j+m+1}-x_j)\sum_{k=j}^{j+m+1}\frac{(x_k-x)_+^m}{\omega'_{m,j}(x_k)} \tag{2.49}$$

由式(2.48)定义的 $n+m$ 个样条函数是线性无关的，所以组成 $S(m,\Delta)$ 的一组基. 这样，对任何定义于区间 $[a,b]$，关于分划 Δ：$a=x_0<x_1<\cdots<x_n=b$ 的任何 m 次样条函数 $S(x)\in S(m,\Delta)$ 都可表示为

$$S(x)=\sum_{j=-m}^{n-1}\alpha_jB_j(x) \tag{2.50}$$

这样，求分划 Δ 上的样条函数 $S(x)$ 的问题，就归结为求表示式(2.50)中的系数 $\alpha_{-m},\alpha_{-m+1},\ \cdots,\ \alpha_{n-1}$ 的问题. 求系数 α_j 的问题一般都归结为解方程组. 例如，已知在点 $x_0,x_1,\cdots,x_n$ 上的函数值 $y_0,\ y_1,\cdots,y_n$，及 x_0,x_n 处的一阶导数值 y'_0 和 y'_n，要求三次样条插值函数 $S(x)$，由式(2.50)，为求系数 α_j，需要求解方程组

$$\begin{cases}\displaystyle\sum_{j=-3}^{n-1}\alpha_jB'_{j,3}(x_0)=y'_0\\ \displaystyle\sum_{j=-3}^{n-1}\alpha_jB_{j,3}(x_i)=y_i\ , \quad (i=0,1,\cdots,n)\\ \displaystyle\sum_{j=-3}^{n-1}\alpha_jB'_{j,3}(x_n)=y'_n\end{cases} \tag{2.51}$$

当求得 $n+3$ 个系数 $\alpha_{-3},\alpha_{-2},\cdots,\alpha_{n-1}$，即得三次样条插值函数 $S(x)$. 为了进一步说明式(2.51)的系数矩阵的特点并研究方程组式(2.51)解的存在唯一性，以及确定系数 α_j，必须了解 B-样条函数的性质.

由 **B**-样条函数的定义，注意到差商的性质，可以得到 **B**-样条函数具有下述重要性质：

(1) 递推关系：

$$B_{j,0}(x)=\begin{cases}1, & x\in(x_j,\ x_{j+1})\\ 0, & \text{其他}\end{cases}$$

$$B_{j,k}(x)=\frac{x-x_j}{x_{j+k}-x_j}B_{j,k-1}(x)+\frac{x_{j+k+1}-x}{x_{j+k+1}-x_{j-1}}B_{j-1,k-1}(x)$$

$$(k=1,2,\cdots,m) \tag{2.52}$$

(2) 正性与局部非零性：

$$B_{j,m}(x)=\begin{cases}0, & x\overline{\in}[x_j,\ x_{j+m+1}]\\ >0, & x\in(x_j,\ x_{j+m+1})\end{cases} \tag{2.53}$$

(3) 规范性：

$$\sum_{j=-m}^{n-1}B_{j,m}(x)=\sum_{j=i-m}^{i}B_{j,m}(x)=1,\quad x_i\leqslant x\leqslant x_{i+1} \tag{2.54}$$

(4) B-样条函数的导数：

当 $m=0$ 时，$B'_{j,0}(x)=0$；

当 $m\geqslant1$ 时($m=1$ 时除 x 为节点外)

$$B'_{j,m}=m\left[\frac{B_{j,m-1}(x)}{x_{j+m}-x_j}-\frac{B_{j+1,m-1}(x)}{x_{j+m+1}-x_{j+1}}\right] \tag{2.55}$$

在样条函数的研究中，无论是理论分析还是实际数值计算，**B**-样条函数都有它的独特作用.

2.7 正交多项式

一、预备知识

1. 权函数的概念

定义 2.8 设 $\rho(x)$定义在有限或无限区间$[a,b]$上，如果具有下列性质：

(1) $\rho(x)\geqslant 0$， 对任意 $x\in[a,b]$；

(2) 积分$\int_a^b x^k\rho(x)\mathrm{d}x$， 对 $k=0,1,2,\cdots$ 都存在；

(3) 对区间$[a,b]$上的非负连续函数 $f(x)$，若

$$\int_a^b f(x)\rho(x)\mathrm{d}x=0,$$

则

$$f(x)\equiv 0$$

则称 $\rho(x)$ 为$[a,b]$上的权函数.

常见的权函数有

$$\rho(x)=1,\ (-1\leqslant x\leqslant 1);\quad \rho(x)=\frac{1}{\sqrt{1-x^2}},(-1\leqslant x\leqslant 1);$$

$$\rho(x)=\mathrm{e}^{-x},\ (0<x<+\infty);\quad \rho(x)=\mathrm{e}^{-x^2},\ (-\infty<x<+\infty).$$

2. 内积的概念

定义 2.9 设 $f(x),g(x)$ 是区间$[a,b]$上的连续函数，$\rho(x)$是$[a,b]$上的权函数，则称

$$(f,g)=\int_a^b\rho(x)f(x)g(x)\mathrm{d}x$$

为函数 $f(x)$ 与 $g(x)$ 在$[a,b]$上以 $\rho(x)$ 为权函数的内积.

内积有如下性质：

(1) $(f,f)\geqslant 0$,当且仅当 $f\equiv 0$ 时,$(f,f)=0$;

(2) $(f,g)=(g,f)$;

(3) $(c_1f_1+c_2f_2,g)=c_1(f_1,g)+c_2(f_2,g)$, $c_1,c_2\in R$.

3. 正交性概念

定义 2.10 若内积

$$(f,g)=\int_a^b\rho(x)f(x)g(x)\mathrm{d}x=0$$

则称 $f(x),g(x)$ 在区间$[a,b]$上带权 $\rho(x)$ 正交. 若函数系 $\{\varphi_0(x),\varphi_1(x),\cdots,\varphi_n(x),\cdots\}$ 满足

$$(\varphi_i,\varphi_j)=\int_a^b\rho(x)\varphi_i(x)\varphi_j(x)\mathrm{d}x=\begin{cases}0, & (i\neq j)\\ A_j>0, & (i=j)\end{cases}$$

则称函数系$\{\varphi_k(x)\}$是$[a,b]$上带权 $\rho(x)$ 的正交函数系. 特别地,当 $A_j\equiv 1$ 时,则称该函数系为标准正交函数系. 若 $\varphi_k(x)(k=0,1,2,\cdots)$ 是最高次项系数不为零的 k 次多项式,则称$\{\varphi_k(x)\}$是$[a,b]$上带权 $\rho(x)$ 的正交多项式系.

4. 函数系的线性关系

定义 2.11 若函数 $\varphi_0(x),\varphi_1(x),\cdots,\varphi_n(x)$,在区间$[a,b]$上连续,如果关系式

$$a_0\varphi_0(x)+a_1\varphi_1(x)+a_2\varphi_2(x)+\cdots+a_n\varphi_n(x)=0$$

当且仅当 $a_0=a_1=a_2=\cdots=a_n=0$ 时才成立,则称函数 $\varphi_0(x)$, $\varphi_1(x),\cdots,\varphi_n(x)$ 在$[a,b]$上是线性无关的,否则称线性相关.

如果函数系$\{\varphi_k(x)\}(k=0,1,2,\cdots)$中任何有限个函数线性无关,则称函数系$\{\varphi_k(x)\}$为线性无关函数系.

例如 $\{1,x,x^2,\cdots,x^n,\cdots\}$ 就是在区间$[a,b]$上的线性无关函数系.

定理 2.7 连续函数 $\varphi_0(x)$, $\varphi_1(x)$, $\cdots,\varphi_n(x)$ 在$[a,b]$上线性无关的充分必要条件是它们的克莱姆行列式 $G_n\neq 0$,

其中

$$G_n = G_n(\varphi_0, \varphi_1, \cdots, \varphi_n) = \begin{vmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \vdots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{vmatrix} \tag{2.56}$$

显然,在区间$[a,b]$上带权$\rho(x)$的正交函数系是线性无关的函数系.

任何线性无关的函数系$\{\varphi_k(x)\}(k=0,1,2,\cdots)$均可通过逐步正交化过程构成正交函数系$\{\psi_k(x)\}(k=0,1,2,\cdots)$,其中$\psi_k(x)$是$\varphi_0(x),\varphi_1(x),\cdots,\varphi_k(x)$的线性组合.如由单项式$\{1,x,\cdots,x^k,\cdots\}$构成的正交多项式可表示为

$$\psi_k(x) = a_k x^k + \cdots + a_1 x + a_0, \quad (a_k \neq 0)$$

它满足条件

$$(\psi_k, \psi_j) = \int_a^b \rho(x)\psi_k(x)\psi_j(x)\mathrm{d}x = \begin{cases} 0\ , & (j \neq k) \\ A_k > 0, & (j = k) \end{cases}$$

其中$\rho(x)$为$[a,b]$上的权函数.当$\rho(x)$和$[a,b]$取得不同,便可得到不同的正交多项式.

二、常用的正交多项式

1. 切比雪夫(Chebyshev)多项式

定义 2.12 称多项式

$$T_n(x) = \cos(n \cdot \mathrm{arc}\cos x), \ (-1 \leqslant x \leqslant 1,\ n = 0,\ 1,\ 2,\ \cdots)$$

为n次的(第一类)切比雪夫多项式.

切比雪夫多项式有以下的重要性质:

(1) **正交性**：$\{T_n(x)\}$ 是在区间 $[-1,1]$ 上带权 $\rho(x)=\frac{1}{\sqrt{1-x^2}}$ 的正交多项式系．即

$$\int_{-1}^{1}\frac{1}{\sqrt{1-x^2}}T_m(x)T_n(x)\mathrm{d}x=\begin{cases}0, & m\neq n\\ \frac{\pi}{2}, & m=n\neq 0\\ \pi, & m=n=0\end{cases} \tag{2.57}$$

(2) **递推关系**：相邻的三个切比雪夫多项式具有三项递推关系式：

$$\begin{cases}T_0(x)=1, \quad T_1(x)=x\\ T_{n+1}(x)=2xT_n(x)-T_{n-1}(x), \quad (n=1,2,\cdots)\end{cases} \tag{2.58}$$

由此可依次写出如下常用的前面几个切比雪夫多项式：

$$\begin{aligned}&T_0(x)=1\\ &T_1(x)=x\\ &T_2(x)=2x^2-1\\ &T_3(x)=4x^3-3x\\ &T_4(x)=8x^4-8x^2+1\\ &T_5(x)=16x^5-20x^3+5x\\ &\cdots\end{aligned} \tag{2.59}$$

(3) **奇偶性**：当 n 为奇数时，$T_n(x)$ 为奇函数；当 n 为偶数时，$T_n(x)$ 为偶函数，从而有

$$T_n(-x)=(-1)^nT_n(x).$$

(4) $T_n(x)$ 在区间 $[-1,1]$ 上有 n 个互异的实零点

$$x_k = \cos\frac{(2k-1)\pi}{2n}, \quad (k=1,2,\cdots,n)$$

2. 勒让德(Legendre) 多项式

定义 2.13　称由

$$\begin{cases} P_0(x) = 1 \\ P_n(x) = \dfrac{1}{2^n \cdot n!} \cdot \dfrac{\mathrm{d}^n}{\mathrm{d}x^n}[(x^2-1)^n], \quad (n=1,2,\cdots) \end{cases} \tag{2.60}$$

确定的 $P_n(x)(n=0,1,2,\cdots)$ **为勒让德多项式.**

勒让德多项式有以下的重要性质:

(1) **正交性:**$\{P_n(x)\}$ **是在区间**$[-1,1]$**上带权** $\rho(x)=1$ **的正交多项式系.即**

$$\int_{-1}^{1} P_m(x)P_n(x)\mathrm{d}x = \begin{cases} 0, & m \neq n \\ \dfrac{2}{2n+1}, & m = n \end{cases} \tag{2.61}$$

(2) **递推关系:相邻的三个勒让德多项式具有三项递推关系式**

$$\begin{cases} P_0(x) = 1, \quad P_1(x) = x \\ P_{n+1}(x) = \dfrac{2n+1}{n+1}xP_n(x) - \dfrac{n}{n+1}P_{n-1}(x), (n=1,2,\cdots) \end{cases} \tag{2.62}$$

由 $P_0(x)=1, P_1(x)=x$,**利用式**(2.62)**就可推出**

$$P_2(x) = \frac{1}{2}(3x^2-1)$$

$$P_3(x) = \frac{1}{2}(5x^3-3x)$$

$$P_4(x) = \frac{1}{8}(35x^4-30x^2+3)$$

$$P_5(x) = \frac{1}{8}(63x^5 - 70x^3 + 15x)$$

$$\cdots \tag{2.63}$$

(3) **奇偶性**:当 n 为奇数时,$P_n(x)$ 为奇函数;当 n 为偶数时,$P_n(x)$ 为偶函数,从而有 $P_n(-x) = (-1)^n P_n(x)$.

(4) $P_n(x)$ 在区间$[-1,1]$上有 n 个互异的实零点.

3. 其他常用的正交多项式

一般而言,如果给定的区间$[a,b]$不同且权函数 $\rho(x)$ 不同,对应的正交多项式也就不同.除了上面两类重要的正交多项式外,再给出两种常用的正交多项式:

(1) **拉盖尔**(Laguerre)**多项式**

定义 2.14 称

$$\boldsymbol{L_n(x)} = e^x \frac{d^n}{dx^n}(x^n e^{-x}), \quad (0 \leqslant x < +\infty), (n = 0,1,2,\cdots) \tag{2.64}$$

为拉盖尔多项式.

可以证明

① $\{L_n(x)\}$ 是在区间$[0,+\infty)$上带权 $\rho(x) = e^{-x}$ 的正交多项式系.即

$$\int_0^{\infty} e^{-x} L_m(x) L_n(x) dx = \begin{cases} 0, & m \neq n \\ (n!)^2, & m = n \end{cases}$$

② 相邻的三项具有递推关系式:

$$\begin{cases} L_0(x) = 1, \quad L_1(x) = 1 - x \\ L_{n+1}(x) = (1 + 2n - x)L_n(x) - n^2 L_{n-1}(x), \quad (n = 1,2,\cdots) \end{cases}$$

(2) **埃尔米特多项式**

定义 2.15 称

$$\boldsymbol{H}_n(\boldsymbol{x}) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2})$$

$$x \in (-\infty, +\infty), \ (n = 0,1,2,\cdots) \tag{2.65}$$

为埃尔米特多项式.

可以证明

① $\{H_n(x)\}$ 是在区间$(-\infty, +\infty)$ 上带权 $\rho(x) = e^{-x^2}$ 的正交多项式系. 即

$$\int_{-\infty}^{+\infty} e^{-x^2} H_m(x) H_n(x) dx = \begin{cases} 0, & m \neq n \\ 2^n n! \sqrt{\pi}, & m = n \end{cases}$$

② 相邻的三项具有递推关系式

$$\begin{cases} H_0(x) = 1, \quad H_1(x) = 2x \\ H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \quad (n = 1,2,\cdots) \end{cases}$$

2.8 最佳平方逼近

用简单函数 $P(x)$ 去近似一个给定在区间$[a,b]$ 上的连续函数 $f(x)$,是函数逼近要研究的问题. 它与插值问题相类似,也是在某一函数类中求函数 $P(x)$,使它与被逼近函数 $f(x)$ 之间满足一定的近似条件. 在插值问题中,这个近似条件就是插值条件,而在逼近问题中,这个近似条件常用逼近函数与被逼近函数之间的某种"距离"来表示. 当用不同的度量标准来定义"距离"时就得到不同的逼近,下面主要讨论比较容易计算的最佳平方逼近.

设 $\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)$ 是$[a,b]$ 上线性无关的连续函数, $a_0, a_1, \cdots, a_n$ 是任意实数,则

$$S(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x)$$

的全体是 $C[a,b]$ 的一个子集,记为

$$\boldsymbol{\Phi} = \mathrm{Span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$$

并称 $\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)$ 是生成集合 $\boldsymbol{\Phi}$ 的一个基底.

例如,$\boldsymbol{P}_n = \mathrm{Span}\{1, x, x^2, \cdots, x^n\}$ 表示由基底 $1, x, \cdots, x^n$ 生成的普通多项式集合.

定义 2.16 对于给定的区间$[a,b]$上的连续函数 $f(x)$,如果存在$S^* \in \boldsymbol{\Phi} = \mathrm{Span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ 使

$$\int_a^b \rho(x)[f(x) - S^*(x)]^2 \mathrm{d}x = \min_{S(x) \in \boldsymbol{\Phi}} \int_a^b \rho(x)[f(x) - S(x)]^2 \mathrm{d}x \tag{2.66}$$

则称 $S^*(x)$是 $f(x)$在集合 $\boldsymbol{\Phi}$ 中的最佳平方逼近函数.

在 $\boldsymbol{\Phi} = P_n = \mathrm{Span}\{1, x, x^2, \cdots, x^n\}$时,满足条件式(2.66)的 $S^*(x)$是 $f(x)$的 n 次最佳平方逼近多项式,简称为 n 次最佳平方逼近.

设 $S(x) = \sum_{j=0}^{n} a_j \varphi_j(x)$. 显然,求最佳平方逼近函数 $S^*(x) = \sum_{j=0}^{n} a_j^* \varphi_j(x)$的问题可归结为求它的系数 $a_0^*, a_1^*, \cdots, a_n^*$,使多元函数

$$I(a_0, a_1, \cdots, a_n) = \int_a^b \rho(x)[f(x) - \sum_{j=0}^{n} a_j \varphi_j(x)]^2 \mathrm{d}x$$

取得最小值,点$(a_0^*, a_1^*, \cdots, a_n^*)$是 $I(a_0, a_1, \cdots, a_n)$的极值点. 由于$I(a_0, a_1, \cdots, a_n)$是关于 $a_0, a_1, \cdots, a_n$ 的二次函数,利用多元函数取得极值的必要条件,有

$$\frac{\partial I}{\partial a_k} = 0, \quad (k = 0, 1, 2, \cdots, n)$$

即

$$\frac{\partial I}{\partial a_k} = 2\int_a^b \rho(x)[f(x) - \sum_{j=0}^{n} a_j \varphi_j(x)][-\varphi_k(x)]\mathrm{d}x = 0 \tag{2.67}$$

于是得

$$\sum_{j=0}^{n} a_j \int_a^b \rho(x)\varphi_k(x)\varphi_j(x)\mathrm{d}x = \int_a^b \rho(x) f(x)\varphi_k(x)\mathrm{d}x$$

$$(k=0,1,\cdots,n) \qquad (2.68)$$

采用函数内积记号

$$(\varphi_k,\varphi_j) = \int_a^b \rho(x)\varphi_k(x)\varphi_j(x)\mathrm{d}x,$$

$$(f,\varphi_k) = \int_a^b \rho(x) f(x)\varphi_k(x)\mathrm{d}x$$

那么,式(2.68)可以简写为

$$\sum_{j=0}^{n}(\varphi_k,\varphi_j)a_j = (f,\varphi_k) \quad , \quad (k=0,1,2,\cdots,n) \qquad (2.69)$$

这是一个包含 $n+1$ 个未知元 $a_0,a_1,\cdots,a_n$ 的 $n+1$ 阶线性代数方程组,写成矩阵形式为

$$\begin{bmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} (f,\varphi_0) \\ (f,\varphi_1) \\ \vdots \\ (f,\varphi_n) \end{bmatrix}$$

$$(2.70)$$

此方程组叫做求 $a_j \quad (j=0,1,2,\cdots,n)$ 的法方程.

显然,其系数行列式就是克莱姆行列式 $G_n = G_n(\varphi_0,\varphi_1,\cdots,\varphi_n)$. 由于 $\varphi_0,\varphi_1,\cdots,\varphi_n$ 线性无关,故 $G_n \neq 0$, 于是方程组(2.70)存在唯一解 $a_k = a_k^*$, $(k=0,1,\cdots,n)$. 从而肯定了函数 $f(x)$ 在 $\boldsymbol{\Phi}$ 中如果存在最佳平方逼近函数,则必是 $S^*(x) = \sum_{j=0}^{n} a_j^* \varphi_j(x)$.

下面进一步证明,由上述方法确定的 $S^*(x)$ 确是函数 $f(x)$ 在 $\boldsymbol{\Phi}$ 中的最佳平方逼近函数. 这只要证明,对任何 $S(x)\in\boldsymbol{\Phi}$,均有

$$\int_a^b \rho(x)[f(x)-S^*(x)]^2\mathrm{d}x \leqslant \int_a^b \rho(x)[f(x)-S(x)]^2\mathrm{d}x$$

因为

$$\int_a^b \rho(x)[f(x)-S(x)]^2\mathrm{d}x-\int_a^b \rho(x)[f(x)-S^*(x)]^2\mathrm{d}x$$

$$=\int_a^b \rho(x)[S(x)-S^*(x)]^2\mathrm{d}x$$

$$+2\int_a^b \rho(x)[S^*(x)-S(x)][f(x)-S^*(x)]\mathrm{d}x$$

其中 $S^*(x)$**的系数** a_k^* **满足方程组**(2.67)**,因此**

$$\int_a^b \rho(x)[f(x)-S^*(x)]\varphi_k(x)\mathrm{d}x=0,\ (k=0,1,2,\cdots,n)$$

而

$$S^*(x)-S(x)=\sum_{k=0}^{n}(a_k^*-a_k)\varphi_k(x)$$

所以

$$\int_a^b \rho(x)[S^*(x)-S(x)][f(x)-S^*(x)]\mathrm{d}x=0$$

于是

$$\int_a^b \rho(x)[f(x)-S(x)]^2\mathrm{d}x-\int_a^b \rho(x)[f(x)-S^*(x)]^2\mathrm{d}x$$

$$=\int_a^b \rho(x)[S(x)-S^*(x)]^2\mathrm{d}x\geqslant 0$$

即

$$\int_a^b \rho(x)[f(x)-S^*(x)]^2\mathrm{d}x\leqslant\int_a^b \rho(x)[f(x)-S(x)]^2\mathrm{d}x$$

这就证明了 $S^*(x)$**确是** $f(x)$**在** Φ **中的最佳平方逼近函数. 以上不仅证明了函数** $f(x)$**在** Φ **中的最佳逼近函数** $S^*(x)$**的存在且唯一,同时也给出了求最佳平方逼近函数** $S^*(x)$**的构造性方法.**

例 2.4 在区间$[0,1]$上给定函数 $f(x)=\sqrt{x}$，求其在

$$\boldsymbol{\Phi}=\mathrm{Span}\{1,x\}$$

中关于 $\rho(x)=1$ 的最佳平方逼近多项式.

解 已知 $\varphi_0(x)=1,\varphi_1(x)=x$ 所求的最佳平方逼近多项式形如

$$P_1(x)=a_0+a_1 x$$

先计算六个内积：

$$(\varphi_0,\varphi_0)=\int_0^1 1^2\mathrm{d}x=1,$$

$$(\varphi_0,\varphi_1)=(\varphi_1,\varphi_0)=\int_0^1 x\mathrm{d}x=\frac{1}{2},$$

$$(\varphi_1,\varphi_1)=\int_0^1 x^2\mathrm{d}x=\frac{1}{3},$$

$$(f,\varphi_0)=\int_0^1 \sqrt{x}\mathrm{d}x=\frac{2}{3},$$

$$(f,\varphi_1)=\int_0^1 x\sqrt{x}\mathrm{d}x=\frac{2}{5},$$

则得 a_0, a_1 应满足的法方程(2.70)为

$$\begin{bmatrix}1 & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{3}\end{bmatrix}\begin{bmatrix}a_0\\ a_1\end{bmatrix}=\begin{bmatrix}\frac{2}{3}\\ \frac{2}{5}\end{bmatrix}\quad \text{解得}\quad \begin{cases}a_0=\dfrac{4}{15}\\ a_1=\dfrac{12}{15}\end{cases}$$

即一次多项式 $P_1(x)=\frac{12}{15}x+\frac{4}{15}$为 $f(x)=\sqrt{x}$在$[0,1]$上的一次最佳平方逼近多项式.

应用中,对于一般的基底 φ_0, φ_1, …, φ_n, 当 n 稍大时,求解

法方程(2.70)的工作量是很大的,若采用 $1, x, x^2, \cdots, x^n$ 作基底,当 $\rho(x)=1$ 时,虽然 $(\varphi_k, \varphi_j)=(x^k, x^j)$ 容易计算,但由此形成的法方程(2.70)的系数矩阵往往是病态的,一般,当 $n\geqslant 4$ 时,其计算结果就不能令人满意.为了克服这个缺点,常用正交基底.其解法如下.

设 $f(x)\in c[a,b]$,$\{\psi_k(x)\}$ 是区间 $[a,b]$ 上带权 $\rho(x)$ 的正交函数系,则在集合 $\psi=\mathrm{Span}\{\psi_0, \psi_1, \cdots, \psi_n\}$ 中求解最佳逼近函数 $S^*(x)=\sum_{k=0}^{n} a_k^* \psi_k(x)$ 的问题,可先求解法方程

$$\sum_{j=0}^{n} (\psi_k, \psi_j) a_j^* = (f, \psi_k), \quad (k=0,1,\cdots,n)$$

因 $\{\psi_k(x)\}$ 是正交函数系,于是法方程可简化为系数矩阵为对角阵的方程组

$$(\psi_k, \psi_k) a_k^* = (f, \psi_k), \quad (k=0, 1, \cdots, n)$$

于是

$$S^*(x)=\sum_{k=0}^{n} \frac{(f, \psi_k)}{(\psi_k, \psi_k)} \psi_k(x) \tag{2.71}$$

若 $\{\psi_k(x)\}$ 还是标准正交的,即进一步有 $(\psi_k, \psi_k)=1$,则式(2.70)更为简单,其系数 $a_k^*=(f, \psi_k)$, $(k=0, 1, \cdots, n)$.

例如,当区间为 $[-1,1]$,$\rho(x)=1$ 时,求区间 $[-1,1]$ 上的连续函数 $f(x)$ 的最小平方逼近时,可以利用勒让德正交多项式 $\{P_n(x)\}$.此时所得的最佳平方逼近多项式为

$$S^*(x)=\sum_{k=0}^{n} a_k^* P_k(x),$$

其中

$$a_k^*=\frac{(f, P_k)}{(P_k, P_k)}=\frac{2k+1}{2}(f, P_k) \quad (k=0,1,\cdots,n) \tag{2.72}$$

例 2.5 求 $f(x)=e^x$ 在$[-1,1]$上的三次最佳平方逼近多项式.

解 先计算四个内积

$$(f,P_0)=\int_{-1}^{1} e^x dx \approx 2.3504,$$

$$(f,P_1)=\int_{-1}^{1} xe^x dx \approx 0.7358,$$

$$(f,P_2)=\int_{-1}^{1} \left(\frac{3}{2}x^2-\frac{1}{2}\right)e^x dx \approx 0.1431,$$

$$(f,P_3)=\int_{-1}^{1} \left(\frac{5}{2}x^3-\frac{3}{2}x\right)e^x dx \approx 0.02013$$

代入式(2.72)得

$$a_0^*=\frac{1}{2}(f,\ P_0)\approx 1.1752,\ a_1^*=\frac{3}{2}(f,\ P_1)\approx 1.1036,$$

$$a_2^*=\frac{5}{2}(f,\ P_2)\approx 0.3578,\ a_3^*=\frac{7}{2}(f,\ P_3)\approx 0.07046$$

于是所求的三次平方逼近多项式为

$$\begin{aligned} S^*(x)&=1.1752\,P_0(x)+1.1036\,P_1(x)\\ &\quad+0.3578\,P_2(x)+0.07046\,P_3(x)\\ &=0.9963+0.9979x+0.5367x^2+0.1761x^3 \end{aligned}$$

如果所给的区间$[a,b]\neq[-1,1]$,应做变换,令

$$x=\frac{a+b}{2}+\frac{b-a}{2}t$$

则$-1\leqslant t\leqslant 1$,于是 $f(x)$变换为 $f\left(\frac{a+b}{2}+\frac{b-a}{2}t\right)$.

例 2.6 求 $f(x)=\sqrt{x}$在区间$[0,1]$上的带权 $\rho(x)=1$ 的一次

最佳平方逼近多项式.

解　令
$$x=\frac{1}{2}(1+t)$$

则
$$f(x)=\frac{1}{\sqrt{2}}\sqrt{1+t}=\varphi(t),\quad -1\leqslant t\leqslant 1$$

先求 $\varphi(t)$在区间$[-1,1]$上的一次最佳平方逼近多项式 $g(t)$,由

$$a_0^*=\frac{1}{2}(\varphi,P_0)=\frac{1}{2}\int_{-1}^{1}\frac{1}{\sqrt{2}}\sqrt{1+t}\mathrm{d}t=\frac{2}{3},$$

$$a_1^*=\frac{3}{2}(\varphi,P_1)=\frac{3}{2}\int_{-1}^{1}\frac{t}{\sqrt{2}}\sqrt{1+t}\mathrm{d}t=\frac{6}{15},$$

可知
$$g(t)=\frac{2}{3}P_0(t)+\frac{6}{15}P_1(t)=\frac{2}{3}+\frac{6}{15}t,\ (-1\leqslant t\leqslant 1)$$

再把 $t=2x-1$ 代入 $g(t)$就得到$\sqrt{x}$在$[0,1]$上的一次最佳平方逼近多项式为

$$P_1(x)=\frac{2}{3}+\frac{6}{15}(2x-1)=\frac{4}{15}+\frac{12}{15}x$$

2.9　曲线拟合的最小二乘法

当函数 $y=f(x)$是由实验或观测得到的,其函数关系通常是由表格$(x_i,\ y_i)(i=1,2,\cdots,m)$形式给出.若用多项式插值方法作表格数据近似,当数据点的数目很大时,要用很高次的多项式,这不仅带来计算上的困难,更主要的缺点是误差很大.若用样条插值函数,虽有很好的数值性质,且计算量也不大,但存放参数 M_i 的量很大,且没有统一的数学公式来表示.另一方面,在有的实际问题中,用插值标准并不合适,当数据点的数目很大时,要求多项式 $P(x)$通过所有数据点,可能失去原函数所表示的规律.如果数据

是由测量得到的，必然带有误差. 而插值法要求 $P(x)$准确通过这些不准确的数据点是不合适的. 在这种情况下，不用插值标准而用近似标准更合理. 因此，怎样从给定的数据表出发，寻找一个简单合理的函数来拟合给定的一组看上去杂乱无章的数据，正是本节要讨论的主要内容. 这里所说的“拟合”，即不要求所作的曲线完全通过所有的数据点，只要求所得的近似曲线能反映数据的基本趋势. 数据拟合在实际中有广泛的应用. 它的实质是在离散情况下的最佳平方逼近，其基本思想和处理方法也具有相似性.

一、一般的最小二乘逼近

设给定数据$(x_i, f_i)\quad(i=1,2,\cdots,m)$，为使逼近函数的构造简单，计算方便，并有良好的逼近性质，可采用最佳平方逼近的作法，即在集合 $\boldsymbol{\Phi}=\mathrm{Span}\{\varphi_0, \varphi_1,\cdots,\varphi_n\}$ 中找一个函数

$$S^*(x)=\sum_{k=0}^{n}a_k^*\varphi_k(x),\ (n<m) \tag{2.73}$$

其误差是

$$\delta_i=S^*(x_i)-f_i,\ (i=1,2,\cdots,m) \tag{2.74}$$

使 $S^*(x)$满足

$$\sum_{i=1}^{m}\omega(x_i)\delta_i^2=\sum_{i=1}^{m}\omega(x_i)[S^*(x_i)-f_i]^2=\min_{S(x)\in\boldsymbol{\Phi}}\sum_{i=1}^{m}\omega(x_i)[S(x_i)-f_i]^2 \tag{2.75}$$

$\omega(x)\geqslant 0$ 是$[a,b]$上给定的权函数. 上述求逼近函数 $S^*(x)$的方法就称为曲线拟合的最小二乘法. 满足关系式(2.75)的函数$S^*(x)$称为上述最小二乘问题的最小二乘解.

曲线拟合的最小二乘解实际上与前面介绍的函数平方逼近相当，只不过它是对离散的变量 $x_1,x_2,\cdots,x_n$ 来求解. 为了求满足条件式(2.75)的最小二乘解

$$S^*(x)=a_0^*\varphi_0+a_1^*\varphi_1+\cdots+a_n^*\varphi_n$$

就要求多元函数

$$I(a_0, a_1, \cdots, a_n) = \sum_{i=1}^{m} \omega(x_i)\left[\sum_{k=0}^{n} a_k \varphi_k(x_i) - f(x_i)\right]^2$$

的极小值点$(a_0^*, a_1^*, \cdots, a_n^*)$，这与 2.8 节讨论的问题完全类似.

由多元函数取极值的必要条件：

$$\frac{\partial I}{\partial a_j} = 0, \ (j = 0, 1, \cdots, n)$$

可得

$$\sum_{i=1}^{m} \omega(x_i)\left\{\sum_{k=0}^{n} a_k \varphi_k(x_i) - f(x_i)\right\}\varphi_j(x_i) = 0$$

若引入离散情况下函数的内积记号：

$$(\varphi_k, \varphi_j) = \sum_{i=1}^{m} \omega(x_i)\varphi_k(x_i)\varphi_j(x_i),$$

$$(f, \varphi_j) = \sum_{i=1}^{m} \omega(x_i) f(x_i)\varphi_j(x_i)$$

则得到法方程

$$\sum_{k=0}^{n} (\varphi_k, \varphi_j) a_k = (f, \varphi_j), \quad (j = 0, 1, 2, \cdots, n) \tag{2.76}$$

它的未知元是 $a_0, a_1, \cdots, a_n$，它的系数是关于 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 的克莱姆行列式，记作 G. 由于 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 线性无关，故 $G \neq 0$，所以方程组(2.76)有唯一解 $a_k = a_k^*$，$(k = 0, 1, \cdots, n)$，从而得到由式(2.73)表达的最小二乘解 $S^*(x)$，它是存在且唯一的.

由上述讨论可以得到如下结论：

(1) 对于给定的函数表$(x_i, f_i)(i = 1, 2, \cdots, m)$，在函数类 $\boldsymbol{\Phi} = \mathrm{Span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$中存在唯一的函数 $S^*(x) = \sum_{k=0}^{n} a_k \varphi_k(x)$使得关系式(2.75)成立；

(2) 最小二乘解的系数 $a_0^*, a_1^*, \cdots, a_n^*$ 可以通过解法方程

(2.76)来获得.

作为曲线拟合的一种常用的情况,如果讨论的是代数多项式拟合,即取

$$\{\varphi_0, \varphi_1, \cdots, \varphi_n\}=\{1, x, x^2, \cdots, x^n\}$$

那么相应的法方程(2.76)就是

$$\begin{bmatrix} \sum \omega_i & \sum \omega_i x_i & \cdots & \sum \omega_i x_i^n \\ \sum \omega_i x_i & \sum \omega_i x_i^2 & \cdots & \sum \omega_i x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum \omega_i x_i^n & \sum \omega_i x_i^{n+1} & \cdots & \sum \omega_i x_i^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum \omega_i f_i \\ \sum \omega_i x_i f_i \\ \vdots \\ \sum \omega_i x_i^n f_i \end{bmatrix} \tag{2.77}$$

其中 $\omega_i=\omega(x_i)$,并且将 $\sum\limits_{i=1}^{m}$ 简写成“$\sum$”.

此时 $S^*(x)=\sum\limits_{k=0}^{n} a_k^* x^k$,称它为数据拟合多项式,上述拟合称为多项式拟合.因为任何连续函数至少在一个比较小的邻域内均可以用多项式任意逼近.因此,在许多实际问题中,不论具体的函数关系如何,都可用适当的多项式作数据拟合.

例 2.7 求数据表

i	1	2	3	4	5
x_i	0	0.25	0.50	0.75	1.00
$f(x_i)$	1.0000	1.2840	1.6487	2.1170	2.7183

的二次最小二乘拟合多项式.

解 设二次拟合多项式为

$$P_2(x)=a_0+a_1x+a_2x^2,$$

将数据表代入式(2.77),得此问题的法方程为

$$
\begin{cases}
5a_0+2.5a_1+1.875a_2=8.7680 \\
2.5a_0+1.875a_1+1.5625a_2=5.4514 \\
1.875a_0+1.5625a_1+1.3828a_2=4.4015
\end{cases}
$$

该方程组的解为 $a_0=1.0052$, $a_1=0.8641$, $a_2=0.8437$.

所以,此数据组的最小二乘拟合二次多项式为

$$P_2(x)=1.0052+0.8641x+0.843x^2$$

以上讨论的都是线性最小二乘问题,即 $S(x)\in\boldsymbol{\Phi}$ **均是形如** $S(x)=\sum_{k=0}^{n}a_k\varphi_k(x)$ **的线性组合. 有的数学模型表面上虽不是线性模型,但可通过变换化为线性模型,则上述方法同样可以使用.**

例 2.8 设给定数据 (x_i,f_i) $(i=1,2,3,4,5)$ **如下:**

x_i	1.00	1.25	1.50	1.75	2.00
$f(x_i)$	5.10	5.79	6.53	7.45	8.46

解 所给数据接近一指数曲线,因而可选择数学模型为指数函数 $y=a\mathrm{e}^{bx}$,(a,b **为待定常数). 可是,这是一个关于** a,b **的非线性模型,为此对** $y=a\mathrm{e}^{bx}$ **两边取对数得** $\ln y=\ln a+bx$, **令** $u=\ln y$, $A=\ln a$, **于是有** $u=A+bx$,**这是一个线性模型,可用最小二乘法求解.**

取 $\varphi_0=1$, $\varphi_1=x$,**要求** $u=A+bx$ **与** (x_i,u_i) $(i=1,2,3,4,5)$ **做最小二乘拟合,由式**(2.77)**得法方程**

$$
\begin{cases}
5A+7.50b=9.404 \\
7.50A+11.875b=14.422
\end{cases}
\quad \text{解得} \quad
\begin{cases}
A=1.122 \\
b=0.5056
\end{cases}
$$

从而 $a=\mathrm{e}^{A}=3.071$,**于是得最小二乘拟合曲线方程为**

$$y=3.071\mathrm{e}^{0.5056x}.$$

若直接用多项式模型 $y=a_0+a_1x+a_2x^2$ 拟合，结果较差.

应该指出，在最小二乘逼近中如何选择数学模型是很重要的，即如何根据给定的数据来选择 Φ，虽然常用的方法是取 $\Phi=\mathrm{Span}\{1,x,\cdots,x^n\}$，但与连续情形的最小平方逼近相似，当 n 较大时，相应的法方程往往也是病态的，因此，当 $n\geqslant 4$ 时，用上述方法计算误差较大，通常要用下面介绍的正交化方法计算. 而对于实际问题中遇到的大量属于 $n\leqslant 3$ 的情形，均可用上面介绍的算法. 目前已有自动选择模型的软件供使用. 应用中，通常要根据物理意义，或数据 (x_i,f_i) 分布的大致图形选择相应的数学模型.

二、用正交函数作最小二乘拟合

为了避免求解病态的法方程，常采用正交函数系作最小二乘拟合.

对于给定的点集 $\{x_i\}$ 和权函数 $\{\omega_i\}(i=1,2,\cdots,m)$，如果函数系 $\varphi_0(x),\varphi_1(x),\cdots,\varphi_n(x)$ 满足

$$(\varphi_j,\varphi_k)=\sum_{i=1}^{m}\omega(x_i)\varphi_j(x_i)\varphi_k(x_i)=\begin{cases}0, & j\neq k\\ A_k>0, & j=k\end{cases}$$

则称函数系 $\{\varphi_j\}_0^n$ 关于点集 $\{x_i\}$ 带权 $\{\omega_i\}$ 正交. 此时，法方程式(2.76)的系数矩阵退化为对角阵，从而由式(2.76)解得

$$a_k^*=\frac{(f,\ \varphi_k)}{(\varphi_k,\varphi_k)}=\frac{\sum_{i=1}^{m}\omega(x_i)f(x_i)\varphi_k(x_i)}{\sum_{i=1}^{m}\omega(x_i)\varphi_k^2(x_i)},\qquad (k=0,1,2,\cdots,n)$$

所以最小二乘解为

$$S^*(x)=\sum_{k=0}^{n}\frac{(f,\ \varphi_k)}{(\varphi_k,\varphi_k)}\varphi_k(x)$$

若已给数据 $(x_i,\ f_i)$ 及权 $\omega_i(i=1,2,\cdots,m)$，可以构造出带权 $\{\omega_i\}$ 的正交多项式 $\{P_k(x)\}$，用递推关系可表示为

$$\begin{cases} P_0(x)=1,\ P_1(x)=x-\alpha_1, \\ P_{k+1}(x)=(x-\alpha_{k+1})P_k(x)-\beta_k P_{k-1}(x), \quad (k=1,2,\cdots,n-1) \end{cases}$$

(2.78)

这里 $P_k(x)$ 是首项系数为 1 的 k 次多项式，α_k 和 β_k 为待定系数，可根据 $\{P_k(x)\}$ 的正交性求得

$$\begin{cases} \alpha_{k+1}=\dfrac{\sum\limits_{i=1}^{m}\omega(x_i)x_iP_k^2(x_i)}{\sum\limits_{i=1}^{m}\omega(x_i)P_k^2(x_i)}=\dfrac{(xP_k,\ P_k)}{(P_k,P_k)} \\ \beta_k=\dfrac{\sum\limits_{i=1}^{m}\omega(x_i)P_k^2(x_i)}{\sum\limits_{i=1}^{m}\omega(x_i)P_{k-1}^2(x_i)}=\dfrac{(P_k,\ P_k)}{(P_{k-1},P_{k-1})}, \quad (k=0,1,\cdots,n-1) \end{cases}$$

(2.79)

与前面讨论类似，此时可得最小二乘逼近多项式为

$$S^*(x)=\sum_{k=0}^{n}a_k^*P_k(x)$$

其中

$$a_k^*=\frac{(f,\ P_k)}{(P_k,\ P_k)}=\frac{\sum\limits_{i=1}^{m}\omega_i f_i P_k(x_i)}{\sum\limits_{i=1}^{m}\omega_i P_k^2(x_i)} \qquad (2.80)$$

用正交化方法求 $S^*(x)$ 时，只要对 $k=0,1,\cdots,n-1$，同时用公式(2.78)，(2.79)及(2.80)计算 $P_k(x)$ 及 a_k^*，其计算简单，又不用解方程组，因此是一个较好的算法.

学习指导

一、基本要求与重点

1．掌握插值与逼近的基本概念与理论，能够按应用问题的不

同情况分别选用插值方法、最佳平方逼近方法和曲线拟合的最小二乘法,去构造符合已知条件的近似函数或计算出某函数的近似函数值,并估计相应的误差.

2. 能熟练地应用构造代数插值多项式的几种常用方法,例如,拉格朗日(Lagrange)插值、牛顿(Newton)插值、分段插值、带导数的埃尔米特(Hermite)插值及三次样条函数插值.

3. 掌握差商与差分的概念,能根据列表数据制作相应的差商表和差分表.

4. 掌握一些常用的正交多项式的概念、性质及其主要应用,例如切比雪夫(Chebyshev)正交多项式、勒让德(Legendre)正交多项式等.

5. 掌握插值法、平方逼近和曲线拟合的最小二乘法的异同点,存在唯一性,并重视各种方法的推导过程、使用条件及应用范围,能根据所给条件建立相应逼近问题的法方程,并会避免和处理病态的法方程.

本章重点:代数插值,最佳平方逼近,曲线拟合的最小二乘法.

二、例题分析与解答

例 1 已知函数 $y=\sin x$ 的函数表如表(例)2-1所示,试分别用拉格朗日线性插值和抛物线插值求 $\sin 0.3367$ 的近似值,并估计截断误差.

表(例)2-1

x	0.32	0.34	0.36
$y=\sin x$	0.314567	0.333487	0.352274

解 (1) 线性插值. 取两个节点 $x_0=0.32$, $x_1=0.34$,相应的 $y_0=0.314567$, $y_1=0.333487$,由公式(2.8)得

$$L_1(x)=l_0(x)y_0+l_1(x)y_1=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_0$$

$$=\frac{x-0.34}{0.32-0.34}\times 0.314567+\frac{x-0.32}{0.34-0.32}\times 0.314567$$

将 $x=0.3367$ 代入,即得

$$\sin 0.3367\approx 0.330365.$$

其截断误差由公式(2.10)得

$$|R_1(x)|\leqslant\frac{M_1}{2}|(x-x_0)(x-x_1)|$$

其中 $M_1=\max\limits_{x_0\leqslant x\leqslant x_1}|f''(x)|$,因 $f(x)=\sin x$, $f''(x)=-\sin x$,可取 $M_1=\max\limits_{x_0\leqslant x\leqslant x_1}|\sin x|=\sin x_1\leqslant 0.3335$,于是

$$|R_1(0.3367)|=|\sin 0.3367-L_1(0.3367)|$$

$$\leqslant\frac{1}{2}\times(0.3335)\times(0.0167)\times(0.0033)$$

$$\leqslant 0.92\times 10^{-5}.$$

(2) 抛物线插值. 取 $x_0=0.32$, $x_1=0.34$, $x_2=0.36$,由公式(2.9)得

$$L_2(x)=l_0(x)y_0+l_1(x)y_1+l_2(x)y_2$$

$$=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1$$

$$+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2$$

将 $x=0.3367$ 和 x_i, $y_i(i=0,1,2)$ 的数值代入,即得

$$\sin 0.3367\approx 0.330374.$$

这个结果与 6 位有效数字的正弦函数表完全一致，这说明查表时用二次插值精度已相当高了. 其截断误差由公式(2.10)得

$$|R_2(x)| \leqslant \frac{M_2}{3!}|(x-x_0)(x-x_1)(x-x_2)|$$

其中 $M_2 = \max\limits_{x_0 \leqslant x \leqslant x_2} |f'''(x)| = \cos x_0 < 0.828.$

于是 $|R_2(0.3367)| = |\sin 0.3367 - L_2(0.3367)|$

$$\leqslant \frac{1}{6} \times (0.828) \times (0.0167)$$

$$\times (0.033) \times (0.0233) < 0.178 \times 10^{-6}.$$

应该指出，由插值的截断误差估计式

$$|R_n(x)| \leqslant \frac{M_n}{(n+1)!}\left|\prod_{j=0}^{n}(x-x_j)\right|$$

可看出，$|R_n(x)|$ 的大小除了与 M_n 有关外，还与因子 $\left|\prod\limits_{j=0}^{n}(x-x_j)\right|$ 有关，为了使 $|R_n(x)|$ 尽量小，显然要尽可能选取靠近插值点 x 的插值节点.

在本题中，若取 $x_1=0.34$，$x_2=0.36$ 为节点作线性插值，可得

$$\sin 0.3367 \approx \widetilde{L}_1(0.3367) = 0.330387$$

其截断误差为 $|\widetilde{R}_2(x)| \leqslant \frac{\widetilde{M}_1}{2}|(x-x_1)(x-x_2)|$

其中 $\widetilde{M}_1 = \max\limits_{x_0 \leqslant x \leqslant x_1} |f''(x)| \leqslant 0.3523.$

于是 $|\widetilde{R}_1(0.3367)| = |\sin 0.3367 - \widetilde{L}_1(0.3367)|$

$$\leqslant \frac{1}{2}(0.3523)(0.0033)(0.0233)$$

$$\leqslant 1.36 \times 10^{-5}.$$

比较以上两种线性插值的节点的不同取法，显然，前者因 $\left|\prod_{j=0}^{n}(x-x_j)\right|$ 比后者小，从而 $|R_1(x)|$ 亦小，相应的插值效果较好．所以，选择插值节点时，应尽量选取与插值点 x 距离较近的一些节点．

例 2 已知 $x=1,2,3,4,5$，对应的函数值为 $f(x)=1,4,7,8,6$．作 4 次牛顿插值多项式．

解 由给定的数据先作差商表如表（例）2-2 所示，以表格中斜线上对应的数字为系数，参照公式（2.14）写出所求的牛顿插值多项式为

表（例）2-2 差商表

k	x_k	$f(x_k)$	一阶差商	二阶差商	三阶差商	四阶差商
0	1	1				
			3			
1	2	4		0		
			3		−1/3	
2	3	7		−1		1/24
			1		−1/6	
3	4	8		−3/2		
			−2			
4	5	6				

$$
\begin{aligned}
N_4(x)&=1+(x-1)\times 3+(x-1)(x-2)\times 0\\
&\quad+(x-1)(x-2)(x-3)\times\left(-\frac{1}{3}\right)\\
&\quad+(x-1)(x-2)(x-3)(x-4)\times\frac{1}{24}\\
&=\frac{1}{24}x^4-\frac{9}{12}x^3+\frac{83}{24}x^2-\frac{33}{12}x+1
\end{aligned}
$$

例 3 已知 $f(x)=\cos x$ 的函数表如表(例)2-3 所示用牛顿向前插值公式计算 $\cos 0.048$ 的值,并估计余项.

表(例)2-3

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6
$\cos x$	1.00000	0.99500	0.98007	0.95534	0.92106	0.87758	0.82534

解 先构造差分表(表(例)2-4),从差分表中可见,四阶差分已接近于常数,用 4 次插值多项式 $N_4(x)\approx f(x)$ 是恰当的. 在计算 $\cos 0.048$ 时,因为 0.048 是靠近表头的值,因此用牛顿向前插值公式计算,这时可取

$$x_0=0.0,\ x_1=0.1,\ x_2=0.2,\ x_3=0.3,\ x_4=0.4,\ h=0.1,$$

表(例)2-4 差分表

x_i	$f(x_i)$	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0.0	1.00000				
		−0.00500			
0.1	0.99500		−0.00993		
		−0.01493		0.00013	
0.2	0.98007		−0.00980		0.00012
		−0.02473		0.00025	
0.3	0.95534		−0.00955		0.00010
		−0.03428		0.00035	
0.4	0.92106		−0.00920		0.00009
		−0.04348		0.00044	
0.5	0.87758		−0.00876		
		−0.05224			
0.6	0.82534				

$$t=\frac{x-x_0}{h}=\frac{0.048-0}{0.1}=0.48$$

公式中用到的各阶差分 $f_0, \Delta f_0, \Delta^2 f_0, \Delta^3 f_0, \Delta^4 f_0$ **正好是表**(**例**) 2-4 **中第一条斜线"——"上的对应值**.

于是

$$\cos 0.048 \approx N_4(0.048)=f_0+\frac{t}{1!}\Delta f_0+\frac{t(t-1)}{2!}\Delta^2 f_0$$

$$+\frac{t(t-1)(t-2)}{3!}\Delta^3 f_0+\frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 f_0$$

$$=1.00000+\frac{0.48}{1!}\times(-0.00500)$$

$$+\frac{0.48(0.48-1)}{2!}\times(-0.00993)$$

$$+\frac{0.48\times(0.48-1)\times(0.48-2)}{3!}\times 0.00013$$

$$+\frac{0.48\times(0.48-1)\times(0.48-2)\times(0.48-3)}{4!}\times 0.00012$$

$$=1-0.0024+0.1248\times 0.00993$$

$$+0.06323\times 0.00013+(-0.03984)$$

$$\times 0.00012\approx 0.99884$$

由余项公式(2.10)**得误差**

$$R_4(x)=\frac{t(t-1)(t-2)(t-3)(t-4)}{5!}h^5 f^{(5)}(\xi),$$

$$(0.0<\xi<0.6)$$

由于 $$|f^{(5)}(\xi)|=|-\sin\xi|\leqslant 1$$

则 $|R_4(x)|\leqslant\left|\dfrac{0.48\times(0.48-1)\times(0.48-2)\times(0.48-3)\times(0.48-4)}{5!}\right.$

$\times 0.1^5 \Big| = 0.28\times 10^{-6}$

例 4 已知 $y=f(x)$ 的函数表如表(例)2-5 所示，求函数 $f(x)$ 在 $[0,2]$ 之间的零点近似值.

表(例)2-5　　函 数 表

x_i	0	1	2
y_i	8	−7.5	−18

解 当函数 $f(x)$ 的解析式未知时，而仅知其在某区间 $[a,b]$ 上的函数表，那么，如何求其在 $[a,b]$ 上的零点呢？一般地，可先求 $f(x)$ 在 $[a,b]$ 上的插值函数 $y(x)$，然后求 $y(x)$ 的零点，把此零点就作为 $f(x)$ 的近似零点. 特别是，若 $f(x)$ 的反函数存在，记为 $x=g(y)$，于是，求 $f(x)$ 的零点问题就变成求函数值 $g(0)$ 的问题了. 若用插值法构造出 $g(y)$，从而求得 $f(x)$ 的零点的近似值，一般称其为反插值，但用反插值时，必须注意反插值的条件是函数 $y=f(x)$ 有反函数，也即要求 $y=f(x)$ 单调. 本题中，因为 y_i 是按严格单调下降排列，所以可用反插值法求 $f(x)$ 的零点的近似值. 具体做法是先把原表变成反函数表(表(例)2-6).

表(例)2-6　　反函数表

y_i	8	−7.5	−18
x_i	0	1	2

由表(例)2-6 构造二次插值多项式 $L_2(y)$，且令 $y=0$，即可得 $f(x)$ 在 $[0,2]$ 之间的零点近似值 $x^*\approx L_2(y)$.

具体计算过程为

$$L_2(0)=\frac{(0+7.5)\times(0+18)}{(8+7.5)\times(8+18)}\times 0$$

$$+\frac{(0-8)(0+18)}{(-7.5-8)(-7.5+18)}\times 1$$

$$+\frac{(0-8)(0+7.5)}{(-18-8)(-18+7.5)}\times 2$$

$$\approx 0.445$$

即 $x^{*}\approx 0.445$.

事实上，此题的 $f(x)=x^{3}-\frac{1}{2}x^{2}-16x+8$ 在$[0,2]$之间的零点为 $x^{*}=0.5$.

应该指出，当所给函数表不满足严格单调的条件时，在利用反插值求根时会出错.

例 5 已知 x_0,x_1,x_2 是$[a,b]$上三个互异的节点，函数 $f(x)$ 在$[a,b]$上具有连续的四阶导数. 试求满足插值条件

$$H(x_i)=f(x_i),\quad (i=0,1,2)$$

$$H'(x_1)=f'(x_1)$$

的插值多项式 $H(x)$，并估计其误差且证明 $H(x)$是唯一的.

解 由给定的 4 个插值条件，显然可确定一个次数不超过 3 次的埃尔米特插值多项式 $H(x)$. 又由 $H(x)$应满足插值条件 $H(x_i)=f(x_i)$，$(i=0,1,2)$，而节点 x_0,x_1,x_2 上的二次插值函数 $N_2(x)$也满足插值条件 $N_2(x_i)=f(x_i)$，$(i=0,1,2)$，故可设$H(x)$的形式为

$$\begin{aligned}H(x)&=N_2(x)+A(x-x_0)(x-x_1)(x-x_2)\\&=f(x_0)+(x-x_0)f[x_0,x_1]\\&\quad+(x-x_0)(x-x_1)f[x_0,x_1,x_2]\\&\quad+A(x-x_0)(x-x_1)(x-x_2)\end{aligned}\tag{2.81}$$

其中 A 为待定系数；显然，由式(2.81)确定的 $H(x)$满足 $H(x_i)$

$=f(x_i),(i=0,1,2)$，常数 A 可由插值条件 $H'(x_1)=f'(x_1)$ 来确定. 为此，对上式两边求导得

$$H'(x)=f[x_0,x_1]+(x-x_1)f[x_0,x_1,x_2]$$
$$+(x-x_0)f[x_0,x_1,x_2]+A[(x-x_1)(x-x_2)$$
$$+(x-x_0)(x-x_2)+(x-x_0)(x-x_1)]$$

令 $x=x_1$，并利用插值条件 $H'(x_1)=f'(x_1)$ 就有

$$f'(x_1)=f[x_0,x_1]+(x_1-x_0)f[x_0,x_1,x_2]$$
$$+A(x_1-x_0)(x_1-x_2)$$

于是

$$A=\frac{f'(x_1)-f[x_0,x_1]-(x_1-x_0)f[x_0,x_1,x_2]}{(x_1-x_0)(x_1-x_2)} \tag{2.82}$$

将式(2.82)代入式(2.81)，即得所求的插值多项式 $H(x)$ 为

$$H(x)=f(x_0)+(x-x_0)f[x_0,x_1]$$
$$+(x-x_0)(x-x_1)f[x_0,x_1,x_2]$$
$$+\frac{f'(x_1)-f[x_0,x_1]-(x_1-x_0)f[x_0,x_1,x_2]}{(x_1-x_0)(x_1-x_2)}$$
$$\cdot(x-x_0)(x-x_1)(x-x_2)$$

再来估计误差

$$R(x)=f(x)-H(x).$$

由插值条件，显然，x_0,x_1,x_2 都是 $R(x)$ 的零点（并且 x_1 是 $R(x)$ 的二重零点），故可设

$$R(x)=k(x)(x-x_0)(x-x_1)^2(x-x_2)$$

其中，$k(x)$ 为待定函数.

为了求得 $k(x)$，暂视 x 为区间 $[a,b]$ 上任意固定的异于 x_i $(i=0,1,2)$ 的点，并作辅助函数

$$g(t)=f(t)-H(t)-k(x)(t-x_0)(t-x_1)^2(t-x_2)$$

显然，$g(t)$ 具有下列性质：

(1) 在 $[a,b]$ 上，$g(t)$ 存在四阶导数，且

$$g^{(4)}(t)=f^{(4)}(t)-k(x)\times 4!$$

(2) $g(x_0)=g(x_1)=g(x_2)=g(x)=g'(x_1)=0$

所以，$g(t)$ 在 (a,b) 内有 5 个零点（其中，x_1 是二重零点，算 2 个），于是类似于余项定理证明，反复应用罗尔定理，最后可得在 $[a,b]$ 内至少有一点 ξ，使

$$g^{(4)}(\xi)=0$$

而
$$g^{(4)}(\xi)=f^{(4)}(\xi)-k(x)\times 4!=0$$

于是

$$k(x)=\frac{1}{4!}f^{(4)}(\xi)$$

从而，余项的表达式为

$$R(x)=\frac{1}{4!}f^{(4)}(\xi)(x-x_0)(x-x_1)^2(x-x_2)$$

其中，ξ 介于 x 与 x_i 之间.

当 $x=x_i$ 时，上式显然也成立.

再证 $H(x)$ 是唯一的（用反证法）.

设另有一个三次多项式 $\widetilde{H}(x)$ 也满足插值条件

$$\widetilde{H}(x_i)=f(x_i)\,(i=0,1,2)\text{ 和 }\widetilde{H}'(x_1)=f'(x_1).$$

于是，函数 $\varphi(x)=H(x)-\widetilde{H}(x)$ 也是次数不超过三次的多项式，且有

$$\varphi(x_i)=H(x_i)-\widetilde{H}(x_i)=f_i-f_i=0,\quad (i=0,1,2)$$

$$\varphi'(x_1)=H'(x_1)-\widetilde{H}(x_1)=f_1'-f_1'=0$$

即 $\varphi(x)$ 至少有 4 个零点，但 $\varphi(x)$ 是次数不超过三次的多项式，所以，只能 $\varphi(x)\equiv 0$，即

$$H(x)\equiv\widetilde{H}(x).$$

唯一性得证.

例 6 已知函数 $y=f(x)$ 的函数表如表(例)2-7 所示，在区间 $[0,0.60]$ 上求三次样条插值函数 $S(x)$，使满足边界条件

表(例)2-7 函数表

i	0	1	2	3	4
x_i	0	0.15	0.30	0.45	0.60
$f(x_i)$	1	0.97800	0.91743	0.83160	0.73529

$$S'(0)=0,\quad S'(0.60)=-0.64879.$$

解 这是在第一种边界条件下的样条插值. 若用三弯矩法求解，则确定 $S''(x_j)=M_j$，$(j=0,1,2,3,4)$ 的方程组为

$$\begin{bmatrix} 2 & 1 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \mu_2 & 2 & \lambda_2 & \\ & & \mu_3 & 2 & \lambda_3 \\ & & & 1 & 2 \end{bmatrix}\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}=\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \tag{2.83}$$

由已知 $h_j=x_j-x_{j-1}=0.15$，得

$$\mu_j=\frac{h_j}{h_j+h_{j+1}}=0.5,\quad \lambda_j=\frac{h_{j+1}}{h_j+h_{j+1}},\ (j=1,2,3)$$

为确定方程组(2.83)的右端项 d_j，$(j=0,1,2,3,4)$，先构造

差商表(表(例)2-8).

表(例)2-8　　差商表

i	x_i	$f(x_i)$	一阶差商	二阶差商
0	0	1		
			−0.14666	
1	0.15	0.97800		−0.85711
			−0.40380	
2	0.30	0.91743		−0.56133
			−0.57220	
3	0.45	0.83160		−0.23288
			−0.64206	
4	0.60	0.73529		

再由公式 $d_j=6f[x_{j-1},x_j,x_{j+1}]$算得各 d_j,($j=0,1,2,3,4$)之值,从而方程组(2.83)为

$$\begin{bmatrix} 2 & 1 & & & \\ 0.5 & 2 & 0.5 & & \\ & 0.5 & 2 & 0.5 & \\ & & 0.5 & 2 & 0.5 \\ & & & 1 & 2 \end{bmatrix}\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}=\begin{bmatrix} -5.86666 \\ -5.14266 \\ -3.36799 \\ -1.39733 \\ -0.26893 \end{bmatrix}$$

解此方程组得

$$M_0=-2.04452,\ M_1=-1.77762,\ M_2=-1.13031,$$

$$M_3=-0.43710,\ M_4=0.08408.$$

代入三次样条插值函数 $S(x)$在各子区间$[x_{j-1},x_j]$上的表达式:

$$S_j(x)=\frac{(x_j-x)^3}{6h_j}M_{j-1}+\frac{(x-x_{j-1})^3}{6h_j}M_j$$

$$+\left(y_{j-1}-\frac{M_{j-1}h_j^2}{6}\right)\frac{x_j-x}{h_j}$$

$$+\left(y_j-\frac{M_j h_j^2}{6}\right)\frac{x-x_{j-1}}{h_j},\ (j=1,2,\cdots,n)$$

(2.88)

经整理得所求的三次样条插值函数 $S(x)$ 在区间 $[0,0.60]$ 上的表达式为

$$S(x)=\begin{cases}0.29655x^3-1.02226x^2+1 & [0,0.15] \\ 0.71923x^3-1.21247x^2+0.02853x+0.99857 & [0.15,0.30] \\ 0.77023x^3-1.25837x^2+0.04230x+0.99720 & [0.30,0.45] \\ 0.57911x^3-1.00035x^2-0.07380x+1.01461 & [0.45,0.60].\end{cases}$$

例 7 在区间 $[0,\pi]$ 上给定函数 $f(x)=\sin x$，求其在 $\boldsymbol{\Phi}=\mathrm{span}\{1,x,x^2\}$ 上的关于 $\rho(x)\equiv 1$ 的最佳平方逼近多项式.

解 已知 $\varphi_0(x)=1,\varphi_1(x)=x,\varphi_2(x)=x^2$，所求的最佳平方逼近多项式形如

$$P_2(x)=a_0+a_1x+a_2x^2$$

先计算 12 个内积：

$$(\varphi_0,\varphi_0)=\int_0^\pi 1\mathrm{d}x=3.141593,$$

$$(\varphi_0,\varphi_1)=(\varphi_1,\varphi_0)=\int_0^\pi x\mathrm{d}x=4.934802,$$

$$(\varphi_0,\varphi_2)=(\varphi_1,\varphi_1)=(\varphi_2,\varphi_0)=\int_0^\pi x^2\mathrm{d}x=10.335426,$$

$$(\varphi_1,\varphi_2)=(\varphi_2,\varphi_1)=\int_0^\pi x^3\mathrm{d}x=24.352273,$$

$$(\varphi_2,\varphi_2)=\int_0^\pi x^4\mathrm{d}x=61.203937,$$

$$(f,\varphi_0)=\int_0^\pi \sin x\mathrm{d}x=2.000006,$$

$$(f,\varphi_1)=\int_0^{\pi} x\sin x\mathrm{d}x=3.141593,$$

$$(f,\varphi_2)=\int_0^{\pi} x^2\sin x\mathrm{d}x=5.869603.$$

则 a_0,a_1,a_2 应满足的法方程式(2.70)为

$$\begin{bmatrix} 3.141593 & 4.934802 & 10.335426 \\ 4.934802 & 10.335426 & 24.352273 \\ 10.335426 & 24.352273 & 61.203937 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}=\begin{bmatrix} 2.000000 \\ 3.141593 \\ 5.869603 \end{bmatrix}$$

解得 $a_0=-0.050454$,$a_1=1.312215$,$a_2=-0.417691$. 即二次多项式 $P_2(x)=-0.050454+1.312215x-0.417691x^2$ 为 $f(x)=\sin x$ 在 $[0,\pi]$ 上的二次最佳平方逼近多项式.

且 $P_2(x)$ 逼近 $f(x)$ 的误差为

$$\|\delta\|_2=\left((f,f)-\sum_{i=0}^{2}a_i(f,\varphi_i)\right)^{\frac{1}{2}}=0.030627.$$

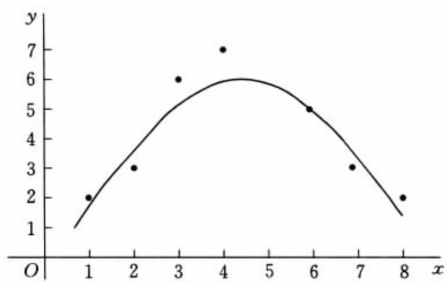
例 8 已知函数 $y=f(x)$ 的实测数据组如表(例)2-9 第二、三列所示,试用最小二乘法求多项式曲线与此数据组拟合.

表(例)2-9 数值表

i	x_i	y_i	x_i^2	x_i^3	x_i^4	x_iy_i	$x_i^2y_i$
1	1	2	1	1	1	2	2
2	2	3	4	8	16	6	12
3	3	6	9	27	81	18	54
4	4	7	16	64	256	28	112
5	6	5	36	216	1296	30	180
6	7	3	49	343	2401	21	147
7	8	2	64	512	4096	16	128
$\sum$	31	28	179	1171	8147	121	635

解 (1) 可先描点:把表中的数据(x_i, y_i)描在坐标纸上,如图(例)2-1 所示. 从图(例)2-1 中可以看出,这些点位于一条抛物线的附近. 因此,可取二次曲线

$$y=a_0+a_1x+a_2x^2$$



图(例)2-1 实测数据点与拟合曲线

作为已给函数 $y=f(x)$的拟合函数.

(2) 建立法方程组(取 $\omega(x_i)=1$)

$$\begin{bmatrix} \sum_{i=1}^{7} 1 & \sum_{i=1}^{7} x_i & \sum_{i=1}^{7} x_i^2 \\ \sum_{i=1}^{7} x_i & \sum_{i=1}^{7} x_i^2 & \sum_{i=1}^{7} x_i^3 \\ \sum_{i=1}^{7} x_i^2 & \sum_{i=1}^{7} x_i^3 & \sum_{i=1}^{7} x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{7} y_i \\ \sum_{i=1}^{7} x_i y_i \\ \sum_{i=1}^{7} x_i^2 y_i \end{bmatrix}$$

由已知数据计算 $\sum_{i=1}^{7} x_i, \sum_{i=1}^{7} x_i^2, \sum_{i=1}^{7} x_i^3, \sum_{i=1}^{7} x_i^4, \sum_{i=1}^{7} y_i, \sum_{i=1}^{7} x_i y_i, \sum_{i=1}^{7} x_i^2 y_i$,计算结果列在表(例)2-9 中.

将(例)表 2-9 中算得的结果代入法方程,即得 a_0, a_1, a_2 应满足的线性代数方程组:

$$\begin{cases} 7a_0+31a_1+179a_2=28 \\ 31a_0+179a_1+1171a_2=121 \\ 179a_0+1171a_1+8147a_2=635 \end{cases}$$

（3）求解法方程得

$$\begin{cases} a_0 = -1.3185 \\ a_1 = 3.4321 \\ a_2 = -0.3864 \end{cases}$$

故所求拟合曲线为

$$y = -1.3185 + 3.4321x - 0.3864x^2$$

它最佳地拟合了表(例)2-9 给出的数据.

例 9 设已知一组实验数据如表(例)2-10 中的第二、三列所示. 试从这组数据出发,建立变量 x 与 y 之间的经验公式.

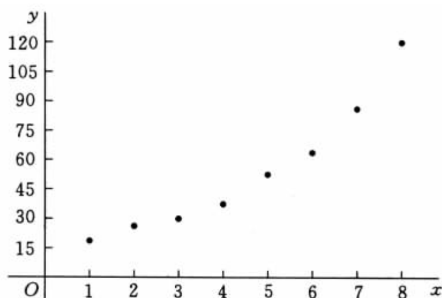
表(例)2-10　　实验数据表

i	x_i	y_i	u_i	x_i^2	x_iu_i
1	1	15.3	1.1847	1	1.1847
2	2	20.5	1.3118	4	2.6236
3	3	27.4	1.4378	9	4.3134
4	4	36.6	1.5635	16	6.2540
5	5	49.1	1.6911	25	8.4555
6	6	65.6	1.8169	36	10.9014
7	7	87.8	1.9435	49	13.6045
8	8	117.6	2.0704	64	16.5632
$\sum_{i=1}^{8}$	36	419.9	13.0197	204	63.9003

解

解题步骤如前,此数据点分布图如图(例)2-2 所示,它接近一指数曲线,因而可取指数函数 $y = ae^{bx}$ (a,b 为待定常数)作为拟合函数.

可是,这是一个关于 a,b 的非线性模型,不是上面讨论的线性



图(例)2-2　实验数据点分布图

无关函数组的线性组合. 因此,首先应通过适当的变换,把 $y=ae^{bx}$ 化为线性模型,然后再利用最小二乘法来求解.

为此,对函数 $y=ae^{bx}$ 两边取常用对数得

$$\lg y=\lg a+bx\lg e,\ 令\ u=\lg y,\ A=\lg a,\ B=b\lg e,$$

则得到了线性模型:

$$u=A+Bx.$$

以下,可类似于例 8 的计算,先列出表(例)2-10 中第四,五,六列数据,再利用数据表(例)2-10,可直接写出法方程:

$$\begin{cases}8A+36B=13.0197\\36A+204B=63.9003\end{cases}$$

求解得

$$A=1.0583,\ B=0.1265$$

再算出 $$a=11.44,\ b=0.2913$$

因此,经验公式为

$$y=11.44e^{0.2912x}$$

从这个例题可知,虽然原来的变量之间不是线性关系,但经过

变量替换后,仍可化为线性关系计算.

例 10 下列数据(表(例)2-11)取自于一个多项式,试确定这个多项式.

表(例)2-11 数据表

x	-2	-1	0	1	2	3
$P(x)$	-43	-13	1	5	5	7

解 由差商的性质知,只要多项式的某阶差商(或差分)均为同一常数,则该多项式的次数必同于该阶差商(或差分)的阶数.

由此,作差商表如表(例)2-12 所示.

表(例)2-12 差商表

x	$f(x)$	差商			
		1	2	3	4
-2	-43				
		30			
			-8		
-1	-13			1	
		14			
0	1		-5		0
		4		1	
1	5		-2		0
		0		1	
2	5		1		
		2			
3	7				

由表(例)2-12 可看出,函数 $f(x)$的三阶差商为常数,$f(x)$应是一个三次多项式,故可从已知数表中任取 4 组数据用牛顿插值或拉格朗日插值构造出该三次多项式为 $P(x)=x^3-5x^2+8x+1$.

例 11 设在区间$[-1,1]$上,要计算 $f(x)=e^x$,欲找一个近似多项式 $P(x)$近似代替 e^x,使误差 $\varepsilon<0.01$.

解 先对 $f(x)=\mathrm{e}^x$ 在 $x=0$ 处作泰勒展开,有

$$\mathrm{e}^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\frac{x^5}{5!}+\cdots$$

若取前六项之和

$$P_5(x)=1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5$$

作为 e^x 的近似,这时截断误差 $R_5(x)$ 满足

$$|R_5(x)|=\left|\frac{1}{6!}\mathrm{e}^{\xi}x^6\right|<\frac{1}{6!}\mathrm{e}=\frac{1}{720}\mathrm{e}<0.0038<0.01$$

即误差已满足所提要求,但注意 $P_5(x)$ 是一个 5 次多项式. 由于 $\{x^k\}$ 也可用切比雪夫多项式 $\{T_n(x)\}$ 表示,即

$$1=T_0,$$

$$x=T_1,$$

$$x^2=\frac{1}{2}(T_0+T_2),$$

$$x^3=\frac{1}{4}(3T_1+T_3),$$

$$x^4=\frac{1}{8}(3T_0+4T_2+T_4),$$

$$x^5=\frac{1}{16}(10T_1+5T_3+T_5),$$

$$\cdots$$

若利用 $\{x^k\}$ 和 $\{T_k(x)\}$ 的关系式,$P_5(x)$ 也可表示为

$$P_5(x)=\frac{81}{64}T_0+\frac{217}{192}T_1+\frac{13}{48}T_2+\frac{17}{384}T_3+\frac{1}{192}T_4+\frac{1}{1920}T_5$$

这里,我们看到 T_k 的足标 k 越大,T_k 的系数就越小,由于在 $|x|\leqslant 1$ 上 $|T_k(x)|\leqslant 1,(k=0,1,2,\cdots)$,故可以略去次数高的 $T_k(x)$ 项,而这就意味着逼近多项式的次数降低,这正是我们希望达到的. 此题中,若略去其最后的两项,则所增添的误差为

$$\frac{1}{192}+\frac{1}{1920}<0.0058$$

从而在 $|x|\leqslant 1$ 上,若用

$$y(x)=\frac{81}{64}T_0+\frac{217}{192}T_1+\frac{13}{48}T_2+\frac{17}{384}T_3$$

近似代替 e^x 的总误差不超过

$$0.0038+0.0058=0.0096<0.01$$

亦即用 $y(x)$ 代替 e^x 也满足所提要求.

再利用 $\{T_k(x)\}$ 和 $\{x^k\}$ 之间的关系,把 $y(x)$ 重写为通常的 x 的乘幂形式,则

$$y(x)=\frac{1}{384}(382+383x+208x^2+68x^3)$$

这个 $y(x)$ 是一个 3 次多项式,比原来的 $P_5(x)$ 降低 2 次,从而用 $y(x)$ 求值的计算工作量比用 $P_5(x)$ 少. 这个方法称为缩减幂级数法.

上述方法既使逼近多项式的次数降低了,而且还可以使误差的分布更为均匀,因此,在应用中也常用切比雪夫多项式 $\{T_n(x)\}$ 来降低逼近多项式的次数.

习　题　二

1. 已知 $f(0)=1$, $f(1)=2$, $f(2)=4$,求 $f(x)$ 的二次插值多项式.

2. 给定函数 $y=\sin x$ 的函数表如下:

x	0.4	0.5	0.6	0.7
$\sin x$	0.389 42	0.479 43	0.564 64	0.644 22

试分别用线性插值与二次插值求 $\sin 0.578\,91$ **的近似值,并估计截断误差.**

3. **设** $l_0(x), l_1(x), \cdots, l_n(x)$ **是以** $x_0, x_1, \cdots, x_n$ **为节点的** n **次基本插值多项式.**

证明

$$\sum_{i=0}^{n} x_i^k \cdot l_i(x) = x^k, \quad (k=0,1,2,\cdots,n).$$

4. **假设对函数** $f(x)$ **在步长为** h **的等距点上造表,且** $|f''(x)| \leqslant M$,**证明:在表中任意相邻两点间做线性插值,误差不超过** $\frac{1}{8}Mh^2$.

设 $f(x)=\sin x$,**问** h **应取多大才能保证线性插值的误差不大于** $\frac{1}{2}\times 10^{-6}$.

5. **已知** $f(x)=x^7+x^4+3x+1$,**求** $f[2^0, 2^1, \cdots, 2^7]$ **及** $f[2^0, 2^1, \cdots, 2^8]$ **的值.**

6. **根据下列函数表,作** 4 **次牛顿插值多项式,并求** $x=1.05$ **时** $\ln x$ **的近似值.**

x	1.00	1.02	1.06	1.08	1.09
$y=\ln x$	0.000 00	0.019 80	0.058 27	0.076 96	0.086 18

7. **已知函数** $y=f(x)$ **的函数表如下:**

x	0.0	0.1	0.2	0.3	0.4	0.5
$f(x)$	1.00	1.32	1.68	2.08	2.52	3.00

试列出相应的向前差分表,并写出牛顿向前插值公式.

8. 试用两种不同的插值方法求一个三次多项式 $P(x)$,使在节点 $x_0=0, x_1=1$ 上满足条件

$$P(0)=f(0)=0, \qquad P(1)=f(1)=1,$$

$$P'(0)=f'(0)=-3, \quad P'(1)=f'(1)=9$$

其中 $P'(x)$, $f'(x)$是 $P(x), f(x)$的一阶导数,并估计余项.

9. 试构造一个四次埃尔米特插值多项式,使其满足下列函数表:

x_i	0	1	2
$f(x_i)$	0	1	1
$f'(x_i)$	0	1	

并估计误差.

10. 已知函数 $y=f(x)$的函数表如下:

x	0	1	4	5
$f(x)$	0	-2	-8	-4

在区间$[0,5]$上求满足边界条件 $S'(0)=\dfrac{5}{2}$, $S'(5)=\dfrac{19}{4}$的三次样条插值函数$S(x)$,并分别计算 $S(x)$在 $x=0.5$, 3, 5 处的值.

11. 若 $f(x)$在$[a,b]$上有三阶连续导数,且已知 $f(x)$在$[a,b]$上两个互异的点 x_0, x_1 上的函数值 $f(x_0), f(x_1)$和一阶导数值 $f'(x_0)$,试利用插值导出$f(x)$的下述表达式:

$$f(x)=-\frac{(x-x_1)(x-2x_0+x_1)}{(x_1-x_0)^2}f(x_0)$$

$$+\frac{(x-x_0)(x-x_1)}{x_0-x_1}f'(x_0)$$

$$+\frac{(x-x_0)^2}{(x_1-x_0)^2}f(x_1)+\frac{1}{6}(x-x_0)^2(x-x_1)f'''(\xi)$$

$$\xi\in(a,b).$$

12. **设** $T_k(x)$**是** k **次切比雪夫多项式**

证明

(1) $T_m[T_n(x)]=T_{mn}(x)$;

(2) $T_{m+n}(x)+T_{m-n}(x)=2T_m(x)T_n(x)$.

13. **求函数** $f(x)=\sqrt{1+x^2}$ **在区间**$[0,1]$**上的一次最佳平方逼近多项式.**

14. **求函数** $f(x)=\sin\pi x$ **在**$[0,1]$**上的二次最佳平方逼近多项式.**

15. **给出数据**

x	-1.00	-0.75	-0.50	-0.25	0	0.25	0.50	0.75	1.00
y	-0.2209	0.3295	0.8826	1.4392	2.0003	2.5645	3.1334	3.7061	4.2836

希望用一次、二次和三次多项式,用最小二乘法拟合这些数据,并写出法方程组.

16. **设有一发射源的发射强度公式为** $I=I_0e^{-\alpha t}$**,现测得** I **与** t **的一组数据如下:**

t_i	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I_i	3.16	2.38	1.75	1.34	1.00	0.74	0.56

试用最小二乘法根据上表确定参数 I_0 **和** α.

第三章　数值积分与数值微分

3.1　数值积分概述

本章主要介绍定积分 $I(f)=\int_a^b f(x)\mathrm{d}x$ 的数值积分方法．

求 $\int_a^b f(x)\mathrm{d}x$ 的值是科学技术中经常遇到的计算问题．对此，虽然在一定条件下，有 Newton-Leibniz 公式

$$\int_a^b f(x)\mathrm{d}x=F(b)-F(a),\quad (F'(x)=f(x))$$

可以计算定积分的值，但是困难的是，在很多情况下，$f(x)$的原函数不易求得，或非常复杂．另外，在一些应用中，函数 $f(x)$是用函数表形式给出而没有解析表达式，这就更无法使用 Newton-Leibniz公式了．因而有必要研究定积分的数值计算方法，以解决定积分的近似计算．

一、求积公式和它的代数精度

由定积分的定义 $\int_a^b f(x)\mathrm{d}x=\lim\limits_{\|\Delta x\|\to 0}\sum\limits_{i=0}^n f(\xi_i)\Delta x_i$ 知，它是和的极限，数值积分就是将定积分的计算用和式近似，它可表示为

$$\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^n \omega_i f(x_i) \tag{3.1}$$

这是用 $f(x)$在节点 x_i 的函数值的某种线性组合来近似定积分，常称为机械求积法．式(3.1)叫做数值求积公式，其中 ω_i 称为求积系数，它是与被积函数 $f(x)$无关的常数，x_i 称为求积节点，

公式中的和

$$Q[f]=\sum_{i=0}^{n}\omega_i f(x_i)$$

称为求积算式，而

$$R[f]=\int_a^b f(x)\mathrm{d}x-\sum_{i=0}^{n}\omega_i f(x_i)$$

称为求积公式的余项．

数值积分的特点是直接用积分区间$[a,b]$上的一些离散节点上的函数值 $f(x_i)$的线性组合计算定积分的近似值，从而将定积分的计算归结为函数值的计算，这就避开了 Newton-Leibniz 公式中需寻求原函数的困难，并为用计算机求积分提供了可行性．但要注意，不应把求积公式(3.1)看作是针对某一特定的函数 $f(x)$，而是要把它看作是$[a,b]$区间上函数整体来讲，所以求积公式也常记为

$$I[f]=Q[f]+R[f]$$

数值求积方法是近似方法，为了研究其精度，常用代数精度这个概念来说明．

定义 3.1 若求积公式(3.1)对所有次数不超过 m 次的代数多项式都精确成立，而对于某个 $m+1$ 次多项式不能精确成立，则称此求积公式具有 m 次代数精度．

求积公式的代数精度概念是衡量公式逼近好坏的标准之一．

一般而言，从上述定义来直接判断求积公式的代数精度比较麻烦．而由多项式和定积分的性质，容易证明上述定义等价于：若依次用 $f(x)=x^0$，x^1，x^2，$\cdots$，x^m 代入求积公式，均有

$$I[f]=Q[f]$$

而用 $f(x)=x^{m+1}$代入求积公式，有

$$I[f]\neq Q[f]$$

例 3.1 设有求积公式

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \omega_0 f(-1) + \omega_1 f(0) + \omega_2 f(1)$$

试确定系数 ω_0，ω_1，ω_2，使上述求积公式的代数精度尽量高，并指出该求积公式所具有的代数精度.

解 令求积公式依次对 $f(x)=1$，x，x^2，都精确成立，即系数 ω_0，ω_1，ω_2 应满足方程组

$$\begin{cases} \omega_0 + \omega_1 + \omega_2 = \int_{-1}^{1} 1\mathrm{d}x = 2 \\ -\omega_0 + \omega_2 = \int_{-1}^{1} x\mathrm{d}x = 0 \\ \omega_0 + \omega_2 = \int_{-1}^{1} x^2\mathrm{d}x = \dfrac{2}{3} \end{cases}$$

解得

$$\omega_0 = \frac{1}{3}，\ \omega_1 = \frac{4}{3}，\ \omega_2 = \frac{1}{3}$$

因此，该求积公式应为

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{1}{3} f(-1) + \frac{4}{3} f(0) + \frac{1}{3} f(1)$$

又容易验证，该求积公式对于 $f(x)=x^3$ 也精确成立，但对 $f(x)=x^4$，求积公式不能精确成立. 因此，该求积公式具有三次代数精度.

二、插值型求积公式

从数值逼近的观点看，对定积分只要能找到一个逼近被积函数足够好，又易于求积分的简单函数 $P(x)$ 近似代替 $f(x)$，从而就有 $\int_a^b f(x)\mathrm{d}x \approx \int_a^b P(x)\mathrm{d}x$. 因此，构造数值求积公式的方法很多，

常用的一种方法是利用插值多项式来构造．

具体做法是，对于定积分 $I[f]=\int_a^b f(x)\mathrm{d}x$，如果已知被积函数在积分区间$[a,b]$上的一组节点

$$a\leqslant x_0<x_1<\cdots<x_n\leqslant b$$

上的函数值 $f(x_0)$，$f(x_1)$，…，$f(x_n)$，
则可构造出 $f(x)$的 n 次拉格朗日插值多项式

$$L_n(x)=\sum_{i=0}^n l_i(x)f(x_i)$$

其中
$$l_i(x)=\prod_{\substack{j=0\\j\neq i}}^n \frac{x-x_j}{x_i-x_j}$$

由于代数多项式的原函数容易求出，则有

$$\int_a^b f(x)\mathrm{d}x\approx\int_a^b L_n(x)\mathrm{d}x=\sum_{i=0}^n f(x_i)\cdot\int_a^b l_i(x)\mathrm{d}x$$

这样就得到了一个数值求积公式

$$\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^n \omega_i f(x_i)$$

其中

$$\omega_i=\int_a^b l_i(x)\mathrm{d}x=\int_a^b \prod_{\substack{j=0\\j\neq i}}^n \frac{x-x_j}{x_i-x_j}\mathrm{d}x \tag{3.2}$$

我们称求积系数由式(3.2)确定的求积公式(3.1)为插值型求积公式．利用插值余项可得插值型求积公式的余项是

$$\begin{aligned}R[f]&=I[f]-Q[f]\\&=\int_a^b [f(x)-L_n(x)]\mathrm{d}x\\&=\int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^n(x-x_j)\mathrm{d}x\end{aligned} \tag{3.3}$$

对于插值型求积公式 $I[f]\approx\sum_{i=0}^{n}\omega_i f_i$，当被积函数 $f(x)$ 取次数不超过 n 次多项式时，由于 $f^{(n+1)}(x)=0$，所以余项 $R[f]\equiv 0$，亦即求积公式对一切次数不超过 n 次的多项式精确成立，所以含有 $n+1$ 个节点 $x_i(i=0,1,2,\cdots,n)$ 的插值型求积公式至少具有 n 次代数精度.

特别地，当 $f(x)\equiv 1$ 时，求积公式也精确成立，从而有 $\sum_{i=0}^{n}\omega_i=b-a$，即插值型求积公式的求积系数之和等于积分区间的长度.

反之，也容易证明，如果一个求积公式 $\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i f(x_i)$ 的代数精度至少是 n 次，那么它可看成是利用插值多项式推导出来的.

3.2 牛顿-柯特斯(Newton-Cotes)求积公式

为了使求积公式的形式简单，下面讨论等距节点下的插值型求积公式.

一、牛顿-柯特斯求积公式的导出

将积分区间$[a,b]$分成 n 等分，记步长 $h=\dfrac{b-a}{n}$，求积节点 x_i 为等距节点 $x_i=a+ih$，$(i=0,1,\cdots,n)$，且对应的函数值 $f(x_i)=f_i$ 为已知，则以这些等距节点为插值节点所导出的插值型求积公式就称为牛顿-柯特斯求积公式，简记为 N-C 公式.

为了进一步简化，作变换 $x=a+sh$，则

$$\mathrm{d}x=h\mathrm{d}s,\ x_i-x_j=(i-j)h,\ x-x_j=(s-j)h,(j=0,1,2,\cdots,n)$$

从而
$$l_i(x)=\prod_{\substack{j=0\\ j\neq i}}^{n}\frac{s-j}{i-j}$$

则 $$\int_a^b l_i(x)\mathrm{d}x = h\int_0^n \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{s-j}{i-j}\mathrm{d}s$$

$$=(b-a)\frac{1}{n}\int_0^n \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{s-j}{i-j}\mathrm{d}s$$

$$=(b-a)\frac{(-1)^{n-i}}{n\cdot i!(n-i)!}\int_0^n \prod_{\substack{j=0\\ j\neq i}}^{n}(s-j)\mathrm{d}s \quad (i=0,1,\cdots,n)$$

记

$$C_i^{(n)}=\frac{(-1)^{n-i}}{n\cdot i!(n-i)!}\int_0^n \prod_{\substack{j=0\\ j\neq i}}^{n}(s-j)\mathrm{d}s,(i=0,1,\cdots,n) \tag{3.4}$$

则等距节点下的插值型求积公式可写成

$$\int_a^b f(x)\mathrm{d}x \approx (b-a)\sum_{i=0}^{n} C_i^{(n)} f(x_i) \tag{3.5}$$

称式(3.5)为 n 阶牛顿-柯特斯求积公式．其中 $C_i^{(n)}$ 称为柯特斯系数，从式(3.4)知 $C_i^{(n)}$ 只与 n,i 有关，而与 $a,b,f(x)$ 都无关，故可事先造好柯特斯系数表，这样使用 N-C 公式计算定积分近似值更加方便．

下面给出 n 从 1～6 的牛顿-柯特斯系数表 3-1.

表 3-1　　牛顿-柯特斯系数表

n	$C_i^{(n)}$						
1	$\frac{1}{2}$	$\frac{1}{2}$					
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$				
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$			
4	$\frac{7}{90}$	$\frac{16}{45}$	$\frac{2}{15}$	$\frac{16}{45}$	$\frac{7}{90}$		
5	$\frac{19}{288}$	$\frac{25}{96}$	$\frac{25}{144}$	$\frac{25}{144}$	$\frac{25}{96}$	$\frac{19}{288}$	
6	$\frac{41}{840}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{34}{105}$	$\frac{9}{280}$	$\frac{9}{35}$	$\frac{41}{840}$

下面给出几个低阶的牛顿-柯特斯公式：

(1) 当 $n=1$ 时，$x_0=a$，$x_1=b$，由公式(3.4)或表 3-1 得

$$C_0^{(1)}=C_1^{(1)}=\frac{1}{2}$$

于是

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{2}[f(a)+f(b)] \tag{3.6}$$

称式(3.6)为梯形公式．常记为

$$T=\frac{b-a}{2}[f(a)+f(b)]$$

(2) 当 $n=2$ 时，$x_0=a$，$x_1=\frac{1}{2}(a+b)$，$x_2=b$，由表 3-1 得

$$C_0^{(2)}=C_2^{(2)}=\frac{1}{6},\ C_1^{(2)}=\frac{4}{6}$$

于是

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right] \tag{3.7}$$

称式(3.7)为抛物线求积公式或辛甫生(Simpson)求积公式．

(3) 当 $n=4$ 时，$x_i=a+i\cdot\frac{b-a}{4}$，$(i=0,1,2,3,4)$，由表 3-1 有

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{90}[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)] \tag{3.8}$$

称式(3.8)为柯特斯公式．

二、牛顿-柯特斯公式的余项

牛顿-柯特斯求积公式是等距节点的插值型求积公式，则当

$f(x)$具有 $n+1$ 阶导数时，其余项也可按式(3.3)表示为

$$R[f]=\int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\mathrm{d}x$$

其中 $\xi\in(a,b)$且依赖于 x.

下面给出上述几个常用的低阶 N-C 公式的余项公式.

定理 3.1 设 $f(x)$在$[a,b]$上有二阶连续导数，则梯形求积公式的余项为

$$R_1[f]=\int_a^b f(x)\mathrm{d}x-\frac{b-a}{2}[f(a)+f(b)]$$

$$=-\frac{(b-a)^3}{12}f''(\eta),\ \eta\in(a,b) \tag{3.9}$$

证明 在 N-C 公式的余项公式(3.3)中，令 $n=1$，可得梯形公式的余项为

$$R_1[f]=\int_a^b \frac{f''(\xi)}{2}(x-a)(x-b)\mathrm{d}x,\ \xi\in(a,b)$$

由于被积函数中$(x-a)(x-b)$在$[a,b]$上恒为负，$f''(\xi)$在$[a,b]$上连续，由积分中值定理，存在 $\eta\in(a,b)$，使

$$R_1[f]=\frac{f''(\eta)}{2}\int_a^b (x-a)(x-b)\mathrm{d}x$$

$$=-\frac{(b-a)^3}{12}f''(\eta)$$

对辛甫生求积公式和柯特斯求积公式，类似的定理成立，由于证明的思想基本类同于定理 3.1，只是具体论证复杂一些，故下面只给出结论.

定理 3.2 若 $f^{(4)}(x)$在$[a,b]$上连续，则抛物线求积公式的余项为

$$R_2[f]=\int_a^b f(x)\mathrm{d}x-\frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)f+f(b)\right]$$

$$=-\frac{1}{2\,880}(b-a)^5 f^{(4)}(\eta),\quad \eta\in(a,b) \tag{3.10}$$

定理 3.3 若 $f^{(6)}(x)$ 在 $[a,b]$ 上连续，则柯特斯求积公式的余项是

$$R_c[f]=-\frac{2(b-a)}{945}\left(\frac{b-a}{4}\right)^6 f^{(6)}(\eta),\eta\in(a,b) \tag{3.11}$$

由式(3.9)、式(3.10)、式(3.11)易知，梯形公式只有一次代数精度，而抛物线求积公式却有三次代数精度，柯特斯公式具有五次代数精度．事实上，由 N-C 公式的余项式(3.3)知，n 阶的 N-C 公式至少有 n 次代数精度，进一步讨论，可以证明当 n 为偶数时，N-C公式对被积函数 $f(x)=x^{n+1}$ 的余项也为零，从而求积公式依然精确成立，所以有以下结论：

定理 3.4 对于 n 阶的牛顿-柯特斯求积公式

$$\int_a^b f(x)\mathrm{d}x\approx(b-a)\sum_{i=0}^{n}C_i^{(n)} f_i$$

当 n 为奇数时，至少具有 n 次代数精度；当 n 为偶数时，至少具有 $n+1$ 次代数精度．

三、牛顿-柯特斯公式的稳定性与收敛性

数值计算的稳定性问题是指初始数据的误差和计算过程中的误差对计算结果的影响大小问题．在数值积分中，由于计算各节点上的函数值 $f(x_i)$ 可能产生误差 δ_i，使实际得到的 $\widetilde{f}(x_i)$ 有误差 $f(x_i)-\widetilde{f}(x_i)=\delta_i$，$(i=0,1,\cdots,n)$. 若记和式 $I[f]=\sum_{i=0}^{n}\omega_i f(x_i)$，$I[\widetilde{f}]=\sum_{i=0}^{n}\omega_i \widetilde{f}(x_i)$，如果其数值积分的计算误差小于给定正数 $\varepsilon(\varepsilon>0)$，即

$$|I[f]-I[\tilde{f}]|=|\sum_{i=0}^{n}\omega_i f_i-\sum_{i=0}^{n}\omega_i \tilde{f}_i|\leqslant\varepsilon \tag{3.12}$$

则表明求积公式的计算是稳定的. 即

定义 3.2 **对于任给的** $\varepsilon>0$,**若存在正数** $\delta>0$,**只要**

$$|f(x_i)-\tilde{f}(x_i)|\leqslant\delta,\quad (i=0,1,\cdots,n)$$

就有式(3.12)**成立,则称求积公式**(3.1)**是稳定的.**

定理 3.5 **若插值型求积公式**(3.1)**中的求积系数** $\omega_i>0$,$(i=0,1,\cdots,n)$,**则此求积公式是稳定的.**

证明 **对任给** $\varepsilon>0$,**若取** $\delta=\dfrac{\varepsilon}{b-a}$,**当** $i=0,1,\cdots,n$ **时,都有** $|f(x_i)-\tilde{f}(x_i)|\leqslant\delta$,**由插值型求积公式有** $\sum\limits_{i=0}^{n}\omega_i=b-a$,**得**

$$|I[f]-I(\tilde{f})|=|\sum_{i=0}^{n}\omega_i[f(x_i)-\tilde{f}(x_i)]|$$

$$\leqslant\sum_{i=0}^{n}|\omega_i||f(x_i)-\tilde{f}(x_i)|$$

$$\leqslant\delta\sum_{i=0}^{n}\omega_i=\frac{\varepsilon}{b-a}(b-a)=\varepsilon$$

由定义 3.2 **知插值型求积公式**(3.1)**当** $\omega_i>0$ **时是稳定的.**

对牛顿-柯特斯公式(3.5),**当** $n\geqslant8$ **时求积系数** $C_i^{(n)}$ **出现负值,因此,公式的稳定性不能保证,在实际应用中,**$n\geqslant8$ **的** N-C **公式不使用.**

定义 3.3 **在求积公式**(3.1)**中,若**

$$\lim_{\substack{n\to\infty\\(h\to0)}}\sum_{i=0}^{n}\omega_i f(x_i)=\int_a^b f(x)\mathrm{d}x$$

其中 $h=\max\limits_{1\leqslant i\leqslant n}(x_i-x_{i-1})$,**则称求积公式**(3.1)**是收敛的.**

可以证明,并非一切连续函数 $f(x)$ **当** $n\to\infty$ **时都有** $R_n[f]\to0$,**即** N-C **公式的收敛性也没有保证. 因此,在实际计算**

时，很少使用高阶的 N-C 公式．

四、复合求积法

由于在实际计算时，不用高阶的 N-C 公式，但若积分区间较大，单独用一个低阶的 N-C 公式来计算积分的近似值，显然精度不好，为了提高数值求积的精度，可利用积分对区间的可加性解决这个问题，这就是通常采用的复合求积法．所谓复合求积法，其指导思想就是先将积分区间分成若干个小区间，在每个小区间上采用低阶求积公式，然后相加得出新的求积公式．下面给出几个常用的低阶复合求积公式．

1．复合梯形求积公式

把区间 n 等分，记分点为 $x_i=a+ih$，$(i=0,1,\cdots,n)$，$h=\dfrac{b-a}{n}$，在每一个小区间$[x_i, x_{i+1}]$上应用梯形公式，就可导出复合梯形公式：

$$\begin{aligned}\int_a^b f(x)\mathrm{d}x &= \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)\mathrm{d}x \\ &\approx \sum_{i=0}^{n-1}\frac{h}{2}[f(x_i)+f(x_{i+1})] \\ &= \frac{h}{2}[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)]\end{aligned}$$

记
$$T_n=\frac{h}{2}[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)] \tag{3.13}$$

这就是复合梯形公式．

由梯形公式的余项式(3.9)得 T_n 的余项为

$$R_T=\int_a^b f(x)-T_n=\sum_{i=1}^{n}\left[-\frac{1}{12}h^3 f''(\eta_i)\right],\ \eta_i\in[x_i, x_{i+1}]$$

设 $f''(x)$在积分区间$[a,b]$上连续，由连续函数性质知，必有

$\eta\in[a,b]$使$f''(\eta)=\frac{1}{n}\sum_{i=1}^{n}f''(\eta_i)$,于是得复合梯形求积公式 T_n 的截断误差为

$$R_T=-\frac{n}{12}h^3f''(\eta)=-\frac{b-a}{12}h^2f''(\eta),\ \eta\in(a,b) \quad (3.14)$$

完全类似地可推导出复合抛物线求积公式及其误差.

2. 复合抛物线求积公式

设 $f^{(4)}(x)$在积分区间$[a,b]$上连续,把$[x_i,\ x_{i+1}]$的中点记作$x_{i+\frac{1}{2}}$,$(i=0,1,\cdots,n-1)$,在每个小区间$[x_i,\ x_{i+1}]$上用抛物线求积公式(3.7),就可推得复合抛物线公式:

$$S_n=\sum_{i=0}^{n-1}\frac{h}{6}[f(x_i)+4f(x_{i+\frac{1}{2}})+f(x_{i+1})]$$

$$=\frac{h}{6}[f(a)+2\sum_{i=1}^{n-1}f(x_i)+4\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}})+f(b)] \quad (3.15)$$

其误差是

$$R_S=\int_a^b f(x)\mathrm{d}x-S_n=-\frac{b-a}{180}\left(\frac{h}{2}\right)^4f^{(4)}(\eta),\ \eta\in(a,b) \quad (3.16)$$

复合抛物线求积公式不但容易编程序上机计算,而且精度也比较高,是一个较好的数值积分法,应用较广泛.

同理,复合柯特斯公式为

$$C_n=\frac{h}{90}[7f(a)+32\sum_{i=0}^{n-1}f(x_{i+\frac{1}{4}})+12\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}})$$

$$+32\sum_{i=0}^{n-1}f(x_{i+\frac{3}{4}})+14\sum_{i=1}^{n-1}f(x_i)+7f(b)] \quad (3.17)$$

其中

$$h=\frac{b-a}{n},\ x_{i+\frac{1}{4}}=x_i+\frac{1}{4}h,\ x_{i+\frac{1}{2}}=x_i+\frac{h}{2},\ x_{i+\frac{3}{4}}=x_i+\frac{3}{4}h$$

其余项是

$$R_C=\int_a^b f(x)\mathrm{d}x-C_n=\frac{-2(b-a)}{945}\left(\frac{h}{4}\right)^6 f^{(6)}(\eta) \tag{3.18}$$

例 3.2 分别用复合梯形公式与复合抛物线公式根据表 3-2 计算积分 $I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$ 的近似值.

表 3-2 已知函数值表

x	0	1/8	1/4	3/8	1/2
$f(x)$	1	0.997 397 8	0.989 615 8	0.976 726 7	0.953 851 0

x	5/8	3/4	7/8	1
$f(x)$	0.936 155 6	0.908 851 6	0.877 192 5	0.841 470 9

解 取 $n=8$,将$[0,1]$8 等分,$h=\frac{1}{8}$,由复合梯形公式(3.13)得

$$\begin{aligned}T_8&=\frac{1}{8}\left[\frac{1}{2}f(0)+f\left(\frac{1}{8}\right)+f\left(\frac{1}{4}\right)+f\left(\frac{3}{8}\right)+f\left(\frac{1}{2}\right)\right.\\&\quad\left.+f\left(\frac{5}{8}\right)+f\left(\frac{3}{4}\right)+f\left(\frac{7}{8}\right)+\frac{1}{2}f(1)\right]\\&\approx 0.945\,690\,9.\end{aligned}$$

若将积分区间 4 等分,用复合抛物线公式(3.15)得

$$\begin{aligned}S_4&=\frac{1}{4\times 6}\left\{f(0)+4\left[f\left(\frac{1}{8}\right)+f\left(\frac{3}{8}\right)+f\left(\frac{5}{8}\right)+f\left(\frac{7}{8}\right)\right]\right.\\&\quad\left.+2\left[f\left(\frac{1}{4}\right)+f\left(\frac{1}{2}\right)+f\left(\frac{3}{4}\right)\right]+f(1)\right\}\\&\approx 0.946\,083\,2.\end{aligned}$$

同积分的精确值 $I=0.946\,083\,1$ 比较,显然复合抛物线求积

法比复合梯形求积法精确得多．一般地说，这个结论在应用中也是对的．

容易证明，以上三个复合求积公式(3.13)、(3.15)和式(3.17)当步长 $h \to 0$ 时均收敛到 $\int_a^b f(x)\mathrm{d}x$，而且收敛速度一个比一个快，并且它们的计算也都是数值稳定的．

五、步长的自动选择

上面介绍的复合求积法对提高积分的精度是行之有效的，但在使用求积公式前，必须给出适当的步长．当 $f(x)$ 在 $[a,b]$ 上的高阶导数容易估计时，利用截断误差式(3.14)或式(3.16)可从预定的精度 ε 估计用复合梯形法或复合抛物线法时的步长 h．但在很多情况下，被积函数 $f(x)$ 的导数界很难估计，这就使得直接预先确定步长 h 有困难．下面介绍一种在计算过程中能自动选取步长的变步长求积法．

"变步长求积法"也叫作区间逐次分半法．其基本思想是：在步长逐次折半过程中，反复用复合求积公式计算，直到二分前后的两次积分计算结果之差的绝对值 $|I_{2n}-I_n|$ 小于允许的精度 ε 为止，并取 I_{2n} 作为所求的积分近似值．这样计算时，步长 h 不是固定不变的，因此叫做变步长求积法．

以复合梯形求积法为例，由式(3.14)有

$$I[f]-T_n=-\frac{b-a}{12}h^2 f''(\eta_1)$$

$$I[f]-T_{2n}=-\frac{b-a}{12}\left(\frac{h}{2}\right)^2 f''(\eta_2)$$

这里 T_n 和 T_{2n} 分别表示将区间 $[a,b]$ n 和 $2n$ 等分的复合梯形公式．若 $f''(x)$ 在 $[a,b]$ 上变化不大，即 $f''(\eta_1)\approx f''(\eta_2)$，并记 $I=I[f]$，则得 $\dfrac{I-T_n}{I-T_{2n}}\approx 4$，经整理，可得

$$I-T_{2n}\approx\frac{1}{3}(T_{2n}-T_n) \tag{3.19}$$

由此可见,当$|T_{2n}-T_n|<3\varepsilon$时,就近似地有$|I-T_{2n}|<\varepsilon$,此时,可取$h=\frac{b-a}{2n}$作为计算复合梯形公式的步长.

按此方法,可以先取$n=1$,即$h=b-a$,然后缩小步长一半,重复这一过程,直到满足精度为止.

为了节省计算量,可建立用T_n来计算T_{2n}的递推式.设

$$h=\frac{b-a}{n},\ x_{i+\frac{1}{2}}=\frac{1}{2}(x_i+x_{i+1}),$$

用复合梯形公式可得

$$\begin{aligned}T_{2n}&=\sum_{i=0}^{n-1}\frac{1}{2}\left(\frac{h}{2}\right)[f(x_i)+2f(x_{i+\frac{1}{2}})+f(x_{i+1})]\\&=\frac{1}{2}\sum_{i=0}^{n-1}\frac{h}{2}[f(x_i)+f(x_{i+1})]+\frac{h}{2}\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}})\\&=\frac{1}{2}T_n+\frac{h}{2}\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}})\end{aligned} \tag{3.20}$$

式(3.20)表明,只要在T_n的基础上,加上对分后新增加的n个分点$x_{i+\frac{1}{2}}$上的函数值的组合就可得到T_{2n},这是一个递推公式,利用式(3.19)和式(3.20)就可构成一个自动选择步长的梯形积分算法.也常称为变步长的梯形法则.

完全类似,由复合的抛物线公式可以构成一个自动选择步长的抛物线积分算法.

例 3.3 用变步长的梯形法求积分$I=\int_0^1\frac{\sin x}{x}\mathrm{d}x$的近似值.

解 先对整个积分区间[0,1]使用梯形公式.

对于函数$f(x)=\frac{\sin x}{x}$,它在$x=0$的值补充定义为$f(0)=1$,而$f(1)=0.841\,470\,9$,根据梯形求积公式计算得

$$T_1=\frac{1}{2}[f(0)+f(1)]=0.9207355$$

将积分区间[0,1]二等分,再计算分点 $x=\frac{1}{2}$处的函数值

$$f\left(\frac{1}{2}\right)=0.9588510$$

然后,由递推公式(3.20)得

$$T_2=\frac{1}{2}T_1+\frac{1}{2}f\left(\frac{1}{2}\right)=0.9397933$$

进一步将积分区间[0,1]四等分,并计算新分点上的函数值,有

$$f\left(\frac{1}{4}\right)=0.9896158,\quad f\left(\frac{3}{4}\right)=0.9088516$$

再利用递推式(3.20)有

$$T_4=\frac{1}{2}T_2+\frac{1}{4}\left[f\left(\frac{1}{4}\right)+f\left(\frac{3}{4}\right)\right]=0.9445135$$

这样不断二分下去,计算结果列成表 3-3(表中 k 代表二分次数,区间等分数 $n=2^k$)

表 3-3　　计算结果

k	0	1	2	3	4	5
T_n	0.9207355	0.9397933	0.9445135	0.9456909	0.9459850	0.9460586
k	6	7	8	9	10	11
T_n	0.9460769	0.9460815	0.9460827	0.9460830	0.9460830	0.9460831

积分 I 的精确值为 0.9460831,用变步长的梯形法在积分区间被二分 11 次时才得到了这个结果.

梯形法的算法虽然简单,但精度较差,收敛速度缓慢,如何提

高收敛速度以节省计算量，自然是一个应该重视的问题．

3.3 自适应积分法

自动选步长的积分法，虽然可以在计算过程中自动选择步长，但这个步长对积分区间的各处都是一致的，这对于有些被积函数是太浪费工作量了．当被积函数在整个积分区间上变化不均匀，即在有的部分区间上变化大，而在另一部分变化小，这时为了既达到计算精度又节省工作量，在变化小的部分可用较大的步长，而对变化大的部分可增加节点，用较小步长，这种根据函数变化情况决定求积节点的方法就称为自适应积分法．它比根据精度要求统一确定步长的方法要节省较多的工作量．因为数值积分的主要计算量是化在对函数值的计算上，因此，若求值的次数越少，整个计算量就越小．自适应积分法就是自动地处理对分子区间的过程，使得用最小量的被积函数求值而达到指定的精确度．这一点由将区间分成不同长度的子区间，并分别在这些子区间上用数值积分公式而达到．下面以自适应辛甫生公式为例，说明自适应积分的方法．

采用逐次将区间二等分的方法，为写法统一，将积分区间 $[a,b]$ 记为 $[a,a+h]$，其中 $h=b-a$ 为区间长度，称原区间为 0 级区间．在区间 $[a,a+h]$ 上对积分 $\int_a^{a+h} f(x)\mathrm{d}x$ 用辛甫生公式(3.7)，把结果记为

$$S_1[a,a+h]=\frac{h}{6}\left[f(a)+4f\left(a+\frac{h}{2}\right)+f(a+h)\right] \tag{3.21}$$

将区间分成两个相等的子区间 $\left[a,a+\frac{h}{2}\right]$ 和 $\left[a+\frac{h}{2},\ a+h\right]$，这两个子区间称为 1 级子区间，其长度为 $\frac{h}{2}$，在每个 1 级子区间上

用辛甫生公式计算积分，然后相加得

$$S_2[a,a+h]=S_1\left[a,\ a+\frac{h}{2}\right]+S_1\left[a+\frac{h}{2},\ a+h\right] \tag{3.22}$$

记
$$I[a,a+h]=\int_a^{a+h} f(x)\mathrm{d}x$$

根据辛甫生公式误差估计式(3.10)有

$$\begin{aligned} I[a,a+h]-S_1[a,a+h] &= -\frac{1}{90}\left(\frac{h}{2}\right)^5 f^{(4)}(\eta_1) \\ &= -\frac{1}{90}\left(\frac{h}{2}\right)^5 f^{(4)}\left(a+\frac{h}{2}\right)+O(h^6) \end{aligned} \tag{3.23}$$

$$\begin{aligned} I[a,a+h]-S_2[a,a+h] &= -\frac{1}{180}\left(\frac{h}{4}\right)^4 f^{(4)}(\eta_2) \\ &= -\frac{1}{1440}\left(\frac{h}{2}\right)^5 f^{(4)}\left(a+\frac{h}{2}\right)+O(h^6) \end{aligned} \tag{3.24}$$

其中 $a<\eta_1,\ \eta_2<a+h$，若忽略 $O(h^6)$，由式(3.23)与式(3.24)相减得

$$S_2[a,a+h]-S_1[a,\ a+h]\approx-\frac{15}{1440}\left(\frac{h}{2}\right)^5 f^{(4)}\left(a+\frac{h}{2}\right)$$

于是有

$$|I[a,a+h]-S_2[a,a+h]|\approx\frac{1}{15}|S_2[a,a+h]-S_1[a,a+h]| \tag{3.25}$$

式(3.25)表明，当 $|S_2-S_1|\leqslant\varepsilon$ 时，有 $|I-S_2|\leqslant\frac{1}{15}\varepsilon$.

再将 1 级子区间中的一个或所有的两个二等分，所得的子区

间称为 2 级子区间，其长度为$\frac{1}{2^2}h$，……，如此继续下去，若最后将区间$[a,a+h]$分成 n 个子区间$[a_i, a_{i+1}]$($i=0,1,\cdots,n-1$)，分点为

$$a=a_0<a_1<\cdots<a_n=b=a+h \tag{3.26}$$

这些子区间的长度可能不同，但当$[a_i,a_{i+1}]$为 r 级子区间，则其长度为 $a_{i+1}-a_i=\frac{h}{2^r}$，即 $a_{r+1}=a_r+\frac{h}{2^r}$，区间记为$\left[a_i, a_i+\frac{h}{2^r}\right]$，实际上，这里区间的划分式(3.26)是根据函数的变化情况而定的，函数变化平缓的地方，子区间大，函数变化急剧的地方，子区间就小．如果记 $I[a,a+h]$的近似值为 $S[a,a+h]$，则当区间分为 n 个子区间时，有

$$S[a,a+h]=\sum_{i=0}^{n-1}S_2[a_i, a_{i+1}]$$

若要求计算 $I[f]$的允许误差为 ε，则需有

$$|I[a,a+h]-S[a,a+h]|\leqslant\varepsilon$$

它可能化为 n 个小区间误差和，即

$$\left|\sum_{i=0}^{n-1}(I[a_i, a_{i+1}]-S_2[a_i, a_{i+1}])\right|\leqslant\varepsilon \tag{3.27}$$

如果取每个 r 级子区间上误差控制为$\frac{1}{2^r}\varepsilon$，即取$[a_i,a_{i+1}]$为 r 级子区间，则当

$$|I[a_i,a_{i+1}]-S_2[a_i, a_{i+1}]|\leqslant\frac{\varepsilon}{2^r} \tag{3.28}$$

时，式(3.27)可以得到满足，而由式(3.25)可知，只要

$$\left|S_2\left[a_i, a_i+\frac{h}{2^r}\right]-S_1\left[a_i, a_i+\frac{h}{2^r}\right]\right|\leqslant 15\cdot\frac{\varepsilon}{2^r} \tag{3.29}$$

则有式(3.28),从而式(3.27)也满足.

因此在逐次二等分区间的过程中,可以根据不等式(3.29)判断是否要将一个 r 级子区间继续分成两个 $r+1$ 级子区间.如果式(3.29)成立,则认为在子区间 $[a_i, a_{i+1}]$ 上已达到计算的精确度,因而可以再考虑与 $[a_i, a_{i+1}]$ 右边相邻的那个子区间;否则将继续分 $[a_i, a_{i+1}]$ 为两个相等的 $r+1$ 级子区间.在实际计算时,式(3.29)的右边经常用 $10 \cdot \frac{\varepsilon}{2^r}$ 代替.

计算步骤是将区间 $[a,b]=[a,a+h]$ 分成两个相等的 1 级子区间 $\left[a,a+\frac{h}{2}\right]$, $\left[a+\frac{h}{2}, a+h\right]$,区间长为 $\frac{h}{2}$,在这两个 1 级子区间上用辛甫生求积公式算得结果 $S_1\left[a,a+\frac{h}{2}\right]$, $S_1\left[a+\frac{h}{2}, a+h\right]$,然后在 1 级区间上计算

$$S_2\left[a,a+\frac{h}{2}\right]=S_1\left[a,a+\frac{h}{2^2}\right]+S_1\left[a+\frac{h}{2^2}, a+\frac{h}{2}\right]$$

在 1 级子区间 $\left[a,a+\frac{h}{2}\right]$ 上比较 $S_1\left[a,a+\frac{h}{2}\right]$ 与 $S_2[a,a+h]$,如果不等式

$$\left|S_1\left[a,a+\frac{h}{2}\right]-S_2\left[a,a+\frac{h}{2}\right]\right| \leqslant 10 \cdot \frac{\varepsilon}{2}$$

成立,则说明在 1 级子区间 $\left[a,a+\frac{h}{2}\right]$ 上已达到要求,然后在下一个 1 级子区间 $\left[a+\frac{h}{2}, a+h\right]$ 重复以上相似步骤逐个判断,直到每个小区间达到式(3.29)要求为止.上述计算在各个子区间上,用非复合的辛甫生公式,将各子区间上的积分加起来,就得到整个区间上积分的近似值.这种方法实质上是一个复合求积公式,只是具有不相等的步长而已.因此,自适应积分公式本质上是一个

不等步长的复合求积公式,它以最小的函数求值而获得满意的结果,因为计算函数值是费时间的,因此在相同的精度下自适应积分法比复合积分公式更优越,可以大大节省计算工作量,但计算过程要比单纯用自动选步长的复合求积公式复杂,然而它容易在计算机上实现,而且它特别适合于振荡函数的数值积分运算.

3.4　龙贝格(Romberg)求积算法

一、数值方法中的加速收敛技巧——外推算法

在数值方法中常常利用一序列 $F_1, F_2, \cdots, F_k, \cdots$ 去逼近准确解 F,然后在理论上给出序列$\{F_k\}$收敛于 F 的误差估计.一个有趣的问题是:能否在截断误差估计的基础上,通过简易的办法,在序列$\{F_k\}$的基础上产生新的序列$\{\widetilde{F}_k\}$,而使$\{\widetilde{F}_k\}$比$\{F_k\}$更快地逼近于准确解 F,这就是数值方法的加速收敛技巧.下面通过例子说明.

设函数 $f(x)$在 $x=0$ 处的值为 $f(0)$,但在很多应用问题中,$f(0)$是无法求得的,只能得到一系列函数值序列:$f(h)$, $f\left(\frac{h}{2}\right)$, $\cdots(h>0)$,而且 h 越小,计算的难度就越大,于是提出这样一个问题,能否通过序列 $f(h)$, $f\left(\frac{h}{2}\right)$,$\cdots$构造出一个新的序列,使它更快地收敛于 $f(0)$呢?在某种条件下,这是办得到的.如利用泰勒展开式

$$f(h)=f(0)+hf'(0)+\frac{1}{2!}h^2 f''(0)+\frac{1}{3!}h^3 f'''(0)+\cdots$$

$$f\left(\frac{h}{2}\right)=f(0)+\frac{h}{2}f'(0)+\frac{1}{2!}\left(\frac{h}{2}\right)^2 f''(0)+\frac{1}{3!}\left(\frac{h}{2}\right)^3 f'''(0)+\cdots$$

如果 $f'(0)\neq 0$,那么 $f(h)$,$f\left(\frac{h}{2}\right)$逼近 $f(0)$的阶都是 $O(h)$. 若令 $f_1(h)=2f\left(\frac{h}{2}\right)-f(h)$,那么当 $f''(0)\neq 0$ 时,$f_1(h)$逼近 $f(0)$的误差阶为 $O(h^2)$,因此序列 $f_1(h)$,$f_1\left(\frac{h}{2}\right)\cdots$就更快地收敛到 $f(0)$.

若再令 $f_2(h)=\dfrac{4f_1\left(\frac{h}{2}\right)-f_1(h)}{3}$,那么当 $f'''(0)\neq 0$ 时,$f_2(h)$逼近 $f(0)$的误差阶为 $O(h^3)$,因此序列 $f_2(h)$,$f_2\left(\frac{h}{2}\right),\cdots$比序列 $f_1(h)$,$f_1\left(\frac{h}{2}\right)\cdots$更快地收敛到 $f(0)$. 还可以从 $f_2(h)$构造出 $f_3(h)$,它们都加速了收敛,这种利用若干已经算出的近似值作适当组合以求更精确的近似值的加速收敛的方法就称为外推算法. 在数值积分中常用的李查逊(Richardson)外推算法如下:

设有一个步长为 h 的函数 $F(h)$去逼近一个常数 F^*,其截断误差为

$$F^*-F(h)=a_1h^{P_1}+a_2h^{P_2}+\cdots+a_kh^{P_k}+\cdots \tag{3.30}$$

其中 $$0<P_1<P_2<\cdots<P_{k-1}<P_k<\cdots$$

a_k,P_k 是与 h 无关的常数,也就是说,$F(h)$逼近 F^* 的误差阶是 h^{P_1}. 现在问,能否利用 $F(h)$构造出一个新的函数 $F_1(h)$,使 $F_1(h)$逼近 F^* 的误差阶比 h^{P_1}更高?事实上,只要令

$$F_1(h)=\frac{F(qh)-q^{P_1}F(h)}{1-q^{P_1}},(1-q^{P_1}\neq 0)$$

那么 $F_1(h)$逼近 F^* 的误差阶是 h^{P_2}.

类似地,令

$$F_2(h)=\frac{F_1(qh)-q^{P_2}F_1(h)}{1-q^{P_2}},(1-q^{P_2}\neq 0)$$

那么 $F_2(h)$**逼近** F^* **的误差阶是** h^{P_3}. **重复这样的做法,可以得到一个算法**

$$\begin{cases}F_1(h)=F(h)\\ F_{m+1}(h)=\dfrac{F_m(qh)-q^{P_m}F_m(h)}{1-q^{P_m}},\ m=1,2,\cdots\end{cases}\tag{3.31}$$

其中,q **为满足** $1-q^{P_m}\neq 0(m=1,2,\cdots)$**的适当正数**.

式(3.31)**称为李查逊外推算法**.

定理 3.6　如果 $F(h)$**逼近** F^* **的截断误差由式**(3.30)**给出,那么,由式**(3.31)**定义的** $F_m(h)$**逼近** F^* **的截断误差为**

$$F^*-F_{m+1}(h)=a_{m+1}^{(m+1)}h^{P_{m+1}}+a_{m+2}^{(m+1)}h^{P_{m+2}}+\cdots$$

其中,$a_k^{(m+1)}(k\geqslant m+1)$**是与** h **无关的常数**.(**证略**)

二、龙贝格求积法

将李查逊外推法应用于复合梯形公式所得的积分近似值序列可得到收敛性很好的数值求积法——龙贝格求积法.

设用复合梯形法计算积分 $I=\int_a^b f(x)\mathrm{d}x$ **的近似值,取步长** $h=\dfrac{b-a}{n}$,**并记** $T_1(h)=T_n$,**则有**

$$I\approx T_1(h)=\frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right]$$

当 $f(x)$**在**$[a,b]$**上充分光滑时,可证用** $T_1(h)$**逼近** I **的截断误差是**

$$I-T_1(h)=a_1h^2+a_2h^4+\cdots+a_kh^{2k}+\cdots$$

其中 a_k 是与 h 无关的常数，以 $q=\frac{1}{2}$，按式(3.31)取序列

$$T_{m+1}(h)=\frac{4^m\cdot T_m\left(\frac{h}{2}\right)-T_m(h)}{4^m-1},\ (m=1,2,\cdots) \tag{3.32}$$

用 $T_{m+1}(h)$来逼近 I 的误差为 $O(h^{2(m+1)})$，这种算法就称为龙贝格算法．

当 $m=1$ 时，按式(3.32)有

$$T_2(h)=\frac{4T_1\left(\frac{h}{2}\right)-T_1(h)}{4-1} \tag{3.33}$$

用 $T_2(h)$逼近 I 的误差为 $O(h^4)$．

因为 $T_1(h)=T_n$，$T_1\left(\frac{h}{2}\right)=T_{2n}$，直接计算有

$$T_2(h)=\frac{4}{3}T_1\left(\frac{h}{2}\right)-\frac{1}{3}T_1(h)$$

$$=\frac{4}{3}T_{2n}-\frac{1}{3}T_n=S_n$$

即
$$S_n=\frac{4T_{2n}-T_n}{4-1} \tag{3.34}$$

S_n 是区间$[a,b]n$ 等分后，在每个小区间上应用抛物线求积公式的结果．这就是说，用复合梯形法二分前后的两个积分近似值 T_n 与 T_{2n}按式(3.34)作这样简单的线性组合就可得精度较高的抛物线法的积分近似值 S_n，从而加速了逼近的效果．公式(3.34)称为梯形加速公式．

当 $m=2$ 时，按式(3.32)有

$$T_3(h)=\frac{4^2T_2\left(\frac{h}{2}\right)-T_2(h)}{4^2-1} \tag{3.35}$$

用 $T_3(h)$ 逼近 I 的误差为 $O(h^6)$.

因为 $T_2(h)=S_n$，$T_2\left(\frac{h}{2}\right)=S_{2n}$，直接计算有

$$T_3(h)=\frac{16}{15}T_2\left(\frac{h}{2}\right)-\frac{1}{15}T_2(h)=\frac{16}{15}S_{2n}-\frac{1}{15}S_n=C_n$$

即有

$$C_n=\frac{4^2 S_{2n}-S_n}{4^2-1} \tag{3.36}$$

C_n 是区间$[a,b]n$ 等分后，在每个小区间上应用柯特斯求积公式得到的具有更高精度的复合柯特斯公式的结果．从而又加速了逼近的效果．公式(3.36)称为抛物线加速公式．

当 $m=3$ 时，按式(3.32)有

$$T_4(h)=\frac{4^3 T_3\left(\frac{h}{2}\right)-T_3(h)}{4^3-1} \tag{3.37}$$

又可写成

$$R_n=\frac{4^3 C_{2n}-C_n}{4^3-1} \tag{3.38}$$

称式(3.38)为龙贝格公式．

用 R_n 逼近 I 的误差是 $O(h^8)$．所以式(3.38)也是柯特斯加速公式．

当 $m\geqslant 4$ 时，按式(3.32)还可以构造出新的求积公式，但其线性组合的两个系数分别为 $\frac{4^m}{4^m-1}$ 及 $\frac{1}{4^m-1}$，显然，第一个系数已接近于 1，而第二个系数绝对值很小，因而这样组合构造出来的新公式与前一个公式计算结果差别不大，反而增加了计算工作量，故计算时只用到式(3.38)为止．上述龙贝格求积算法是在变步长的求积过程中运用三个加速公式(3.34)、(3.36)和式(3.38)将变步长的梯形法得到的粗糙的积分近似值迅速加工成精度较高的积分近似值的求积方法．显然龙贝格求积公式(3.38)已经不属于 N-C

公式的范围．龙贝格求积算法是一种区间逐次分半的线性加速收敛方法，是一种外推算法．

三、计算步骤及数值例子

在计算机上应用龙贝格数值积分方法求积分 $I=\int_a^b f(x)\mathrm{d}x$ 的计算步骤如下：

（1）准备初值：先用梯形公式计算积分近似值

$$T_1=\frac{b-a}{2}[f(a)+f(b)]$$

（2）按变步长梯形法则计算积分近似值：将区间逐次分半，令区间长度

$$h=\frac{b-a}{2^i},(i=0,1,2,\cdots)$$

计算 $$T_{2n}=\frac{1}{2}T_n+\frac{h}{2}\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}}),\quad (n=2^i)$$

（3）用三个加速公式求加速值：

梯形加速公式 $$S_n=T_{2n}+\frac{1}{3}(T_{2n}-T_n),$$

抛物线加速公式 $$C_n=S_{2n}+\frac{1}{15}(S_{2n}-S_n),$$

龙贝格求积公式 $$R_n=C_{2n}+\frac{1}{63}(C_{2n}-C_n).$$

（4）精度控制：直到前后两个积分值 R_{2n} 和 R_n 满足如下关系：当 $|R_{2n}|\leqslant 1$，满足（绝对误差）$|R_{2n}-R_n|<\varepsilon$；

当 $|R_{2n}|>1$，满足（相对误差）$\left|\frac{R_{2n}-R_n}{R_{2n}}\right|<\varepsilon$.

则终止计算并取 R_{2n} 作为积分 $\int_a^b f(x)\mathrm{d}x$ 的近似值，否则将区间再

对分，重复(2)，(3)，(4)的计算步骤，直到满足精度为止(ε 为允许的误差限).

例 3.4 用龙贝格求积算法求积分 $I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$ 的近似值，使误差不超过$\frac{1}{2}\cdot 10^{-6}$.

解 按上述步骤计算结果列于表 3-4.

表 3-4 计算结果

k	T_{2^k}	S_{2^k-1}	C_{2^k-2}	R_{2^k-3}
0	0.920 735 5			
1	0.939 793 3	0.946 145 9		
2	0.944 513 5	0.946 086 9	0.946 083 0	
3	0.945 690 9	0.946 083 3	0.946 083 1	0.946 083 1

表 3-4 所列的计算结果表明，只运用了三次二分的复合梯形计算的值，尽管这些近似值的精度都不高，只有两位准确数字，但经过三次利用加速公式所得到的龙贝格积分值 $R_1=0.946\ 083\ 1$，它的每一位数字已经都是准确数字，可见加速的效果是很明显的.

应用龙贝格求积算法，系数有规律，不需要存贮求积系数，占用存贮单元少，收敛速度快，精度又较高，因此很适合在计算机上应用. 它的主要缺点是，每当把区间对分后，就要计算被积函数 $f(x)$在新分点处的值，而这些值的个数又是成倍地增加的，所以计算量较大.

3.5 高斯(Gauss)求积方法

一、高斯求积公式

本节讨论怎样导出形如式(3.1)

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)$$

的求积公式，使它具有最高的代数精度.

我们先看一个例子

例 3.5 求形如

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \omega_0 f(x_0) + \omega_1 f(x_1)$$

的两点求积公式.

解　本题的解法很多，结果也不一定相同，下面介绍两种解法：

（1）用梯形公式（即以 $x_0=-1$，$x_1=1$ 为节点的插值型求积公式）立即可得

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f(-1) + f(1)$$

显然，此求积公式只具有一次代数精度.

（2）若适当选取求积公式中的 4 个待定系数 ω_0，ω_1，x_0，x_1，使求积公式对 $f(x)=1$，x，x^2，x^3 都准确成立，即得方程组

$$\begin{cases} \omega_0 + \omega_1 = 2 \\ \omega_0 x_0 + \omega_1 x_1 = 0 \\ \omega_0 x_0^2 + \omega_1 x_1^2 = \dfrac{2}{3} \\ \omega_0 x_0^3 + \omega_1 x_1^3 = 0 \end{cases}$$

求得

$$\omega_0 = \omega_1 = 1,\ x_0 = -\frac{\sqrt{3}}{3},\ x_1 = \frac{\sqrt{3}}{3}.$$

故得

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f\left(-\frac{\sqrt{3}}{3}\right)+f\left(\frac{\sqrt{3}}{3}\right)$$

显然，此求积公式具有三次代数精度．容易验证，它也是插值型的求积公式．

此例告诉我们，对于两点求积公式，只要适当选择求积节点 x_0，x_1 和求积系数 ω_0，ω_1，可使其代数精度达到三次．由此可见，求积公式的代数精度不仅与积分的节点数有关，而且与这些点的分布有关．适当调整这些点的分布和求积系数，能使求积公式达到最高的代数精度．

在前面讨论牛顿-柯特斯求积公式时，为了方便，把求积节点取作等距，因而求积公式的代数精度也受到了限制，现在问，在一般的求积公式 $\int_a^b f(x)\mathrm{d}x \approx \sum\limits_{i=0}^{n}\omega_i f(x_i)$ 中，若求积节点的数目仍规定为 $n+1$ 个，而让节点的位置和求积系数都能适当选择，能否使求积公式(3.1)的代数精度达到最高呢？下面来讨论常用的插值型求积公式．

由前面的讨论已知，过 $n+1$ 个节点的插值型求积公式至少具有 n 次代数精度，那么其最高代数精度能达到多少呢？

定理 3.7 形如式(3.1)的插值型求积公式的代数精度最高不超过 $2n+1$ 次．

证明 令 $f(x)=(x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2=[\pi(x)]^2$，则 $f(x)$ 为 $2n+2$ 次多项式，且只在 $x_0,x_1,\cdots,x_n$ 处 $f(x)$ 为零，故

$$\int_a^b f(x)\mathrm{d}x=\int_a^b [\pi(x)]^2\mathrm{d}x>0$$

而

$$\sum_{i=0}^{n}\omega_i f(x_i)=0$$

这表明求积公式(3.1)不管系数 ω_i 和节点 x_i 如何选取，它对 $2n+2$次的多项式 $f(x)=[\pi(x)]^2$ 不可能精确成立，因此形如式(3.1)的求积公式的代数精度不能超过 $2n+1$．即形如式(3.1)的

插值型求积公式的最高代数精度不可能大于 $2n+1$，那么最高代数精度为 $2n+1$ 的求积公式存在吗？回答是肯定的，只要适当选择求积节点，可使插值型求积公式的代数精度达到最高，这就是本节将要介绍的高斯求积公式.

定义 3.4 若一组节点 $x_0, x_1, \cdots, x_n \in [a,b]$，使插值型求积公式(3.1)具有最高代数精度 $2n+1$ 次，则称此组节点为高斯点，相应的求积公式称为高斯求积公式.

因为在插值型求积公式中，只要节点 x_i 确定了，系数 ω_i 也随之确定，因此构造高斯求积公式的关键是求高斯点，下面利用正交多项式的特性来构造高斯求积公式.

定理 3.8 插值型求积公式(3.1)中，求积节点 $x_i(i=0,1,\cdots,n)$ 是高斯点的充分必要条件是：在区间 $[a,b]$ 上以这组节点为零点的 $n+1$ 次多项式

$$\pi(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$$

与所有的次数不超过 n 次的多项式 $P(x)$ 都正交，即

$$\int_a^b P(x)\pi(x)\mathrm{d}x=0 \tag{3.39}$$

证明 (1) 必要性：设 $P(x)$ 是任意一个次数不超过 n 次的多项式，则 $P(x)\pi(x)$ 是次数不超过 $2n+1$ 次的多项式，因此，如果 $x_0, x_1, \cdots, x_n$ 是插值型求积公式(3.1)的高斯点，则求积公式对于 $f(x)=P(x)\pi(x)$ 能精确成立，即有

$$\int_a^b P(x)\pi(x)\mathrm{d}x=\sum_{i=0}^{n}\omega_i P(x_i)\pi(x_i)$$

但
$$\pi(x_i)=0,(i=0,1,\cdots,n)$$

所以 $\int_a^b P(x)\pi(x)\mathrm{d}x=0$ 成立.

(2) 充分性：设 $f(x)$ 是任意给定的次数不超过 $2n+1$ 次的多

项式，则有 $f(x)=P(x)\pi(x)+q(x)$，其中 $P(x)$ 和 $q(x)$ 均为次数不超过 n 次的多项式．于是有

$$\int_a^b f(x)\mathrm{d}x=\int_a^b P(x)\pi(x)\mathrm{d}x+\int_a^b q(x)\mathrm{d}x$$

由定理的假设(3.39)得

$$\int_a^b P(x)\pi(x)\mathrm{d}x=0$$

故
$$\int_a^b f(x)\mathrm{d}x=\int_a^b q(x)\mathrm{d}x$$

由于所给的求积公式(3.1)是插值型的求积公式，则它至少具有 n 次代数精度，从而对次数不超过 n 次的多项式 $q(x)$ 能精确成立，即 $\int_a^b q(x)\mathrm{d}x=\sum\limits_{i=0}^{n}\omega_i q(x_i)$，再注意到 $\pi(x_i)=0,(i=0,1,\cdots,n)$，所以

$$f(x_i)=P(x_i)\pi(x_i)+q(x_i)=q(x_i)$$

从而有
$$\int_a^b q(x)\mathrm{d}x=\sum_{i=0}^{n}\omega_i f(x_i)$$

于是
$$\int_a^b f(x)\mathrm{d}x=\sum_{i=0}^{n}\omega_i f(x_i)$$

可见求积公式(3.1)对于一切次数不超过 $2n+1$ 次的多项式都能精确成立，所以它是高斯求积公式，其节点 $x_0,x_1,\cdots,x_n$ 就是高斯点．证毕．

由于 $[a,b]$ 上任意 $n+1$ 次正交多项式都存在，且 $n+1$ 个实的、互异的零点全在 $[a,b]$ 中，故得

推论 3.1 $[a,b]$ 上的 $n+1$ 次正交多项式的 $n+1$ 个零点就是高斯求积公式的高斯点．

高斯求积公式在节点 $x_i(i=0,1,\cdots,n)$ 求出后，求积系数 ω_i $(i=0,1,\cdots,n)$ 可利用代数精度的定义所得的方程组解之，也可直

接由插值求积公式的系数表达式 $\omega_i=\int_a^b l_i(x)\mathrm{d}x$ 得到．

由于勒让德正交多项式是$[-1,1]$上带权 $\rho(x)=1$ 的正交多项式，因此当区间$[a,b]=[-1,1]$时，只要求出 $n+1$ 次的勒让德正交多项式 $P_{n+1}(x)$的 $n+1$ 个零点，就能构造出相应的高斯求积公式．由勒让德正交多项式构造出来的高斯求积公式称为高斯-勒让德公式，也简称为高斯求积公式．它可表示为

$$\int_{-1}^{1} f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i f(x_i) \tag{3.40}$$

其中，节点 $x_i(i=0,1,\cdots,n)$是 $n+1$ 次勒让德正交多项式$P_{n+1}(x)$的$n+1$个零点．

例如：① 一个节点时，因一次勒让德正交多项式 $P_1(x)=x$，其零点为 $x_0=0$，以 $x_0=0$ 为节点构造形如

$$\int_{-1}^{1} f(x)\mathrm{d}x\approx\omega_0 f(0)$$

的高斯求积公式，因求积公式只具有一次代数精度，所以令它对$f(x)=1$精确成立，可求得 $\omega_0=2$，这样构造出来的高斯-勒让德求积公式为

$$\int_{-1}^{1} f(x)\mathrm{d}x\approx 2f(0) \tag{3.41}$$

这就是大家熟悉的中矩形公式．

② 两个节点时，因二次勒让德多项式 $P_2(x)=\frac{1}{2}(3x^2-1)$，其零点为 $x_0=-\frac{1}{\sqrt{3}}$，$x_1=\frac{1}{\sqrt{3}}$，以 $x_0=-\frac{1}{\sqrt{3}}$，$x_1=\frac{1}{\sqrt{3}}$为节点构造形如

$$\int_{-1}^{1} f(x)\mathrm{d}x\approx\omega_0 f\left(-\frac{1}{\sqrt{3}}\right)+\omega_1 f\left(\frac{1}{\sqrt{3}}\right)$$

的高斯求积公式，由于它的代数精度为 3，所以它对 $f(x)=1,x$ 都精确成立，可得

$$\begin{cases} \omega_0 + \omega_1 = 2 \\ \omega_0 \left(-\frac{1}{\sqrt{3}}\right) + \omega_1 \left(\frac{1}{\sqrt{3}}\right) = 0 \end{cases} \quad \text{解得 } \omega_0 = \omega_1 = 1$$

从而得到两点的高斯-勒让德求积公式为

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \tag{3.42}$$

完全类似地,可求出三点的高斯-勒让德求积公式为

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{5}{9} f\left(-\frac{\sqrt{15}}{5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\frac{\sqrt{15}}{5}\right) \tag{3.43}$$

由于求积节点和系数与 $f(x)$ 无关,为了便于应用,对于不同的 n,高斯-勒让德公式中的节点和系数已被制成表,使用时,只要查表就可以方便地写出相应的各个节点的高斯-勒让德求积公式.表 3-5 给出了 1～6 个节点时高斯-勒让德公式的求积节点和系数.

表 3-5　　求积节点和系数表

节点数 n	节　点 x_i	系　数 ω_i
1	0.000 000 0	2.000 000 0
2	±0.577 350 3	1.000 000 0
3	±0.774 596 7 0.000 000 0	0.555 555 6 0.888 888 9
4	±0.861 136 3 ±0.339 981 0	0.347 854 8 0.652 145 2
5	±0.906 179 8 ±0.538 469 3 0.000 000 0	0.236 926 9 0.478 628 7 0.568 888 9
6	±0.932 469 51 ±0.661 209 39 ±0.238 619 19	0.171 324 49 0.360 761 57 0.467 913 93

对于一般区间$[a,b]$应用高斯求积公式时，可先用变量置换：

$$x=\frac{1}{2}[(a+b)+(b-a)t]$$

将它转化为$[-1,1]$上的积分：

$$\int_a^b f(x)\mathrm{d}x=\frac{b-a}{2}\int_{-1}^1 f\left(\frac{1}{2}(a+b)+\frac{1}{2}(b-a)t\right)\mathrm{d}t$$

然后用高斯-勒让德求积公式来计算它，于是得到 $n+1$ 个节点的高斯求积公式：

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{2}\sum_{i=0}^n\omega_i f\left(\frac{1}{2}(a+b)+\frac{1}{2}(b-a)t_i\right) \tag{3.44}$$

其中 ω_i 与 t_i 都有表可查.

例 3.6 用高斯-勒让德公式计算积分 $I=\int_0^1\frac{\sin x}{x}\mathrm{d}x$.

解 先作变换 $x=\frac{1}{2}(t+1)$，把积分区间$[0,1]$化为区间$[-1,1]$，得

$$\int_0^1\frac{\sin x}{x}\mathrm{d}x=\int_{-1}^1\frac{\sin\frac{1}{2}(t+1)}{t+1}\mathrm{d}t$$

用两个节点的高斯-勒让德求积公式有

$$I\approx\frac{\sin\frac{1}{2}(-0.5773503+1)}{-0.5773503+1}+\frac{\sin\frac{1}{2}(0.5773503+1)}{0.5773503+1}$$

$$=0.9460411$$

类似地，用三个节点的高斯-勒让德求积公式得

$$I\approx0.9460831$$

与前面用 N-C 求积公式的计算结果相比较，显然其结果精确得

多，这说明高斯求积公式的精度是很高的．

二、高斯求积公式的余项

利用埃尔米特插值和高斯求积公式的性质，可以推导出高斯求积公式的余项．

定理 3.9 若 $f^{(2n+2)}(x)$ 在区间$[a,b]$上连续，则高斯求积公式的余项

$$R[f]=\int_a^b f(x)\mathrm{d}x-\sum_{i=0}^{n}\omega_i f(x_i)$$

$$=\frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_a^b [\pi(x)]^2\mathrm{d}x \qquad (\eta\in[a,b]) \qquad (3.45)$$

证明 以 $x_0,x_1,\cdots,x_n$ 为节点构造 $f(x)$的次数不超过 $2n+1$ 次的埃尔米特插值多项式 $H(x)$，使满足条件

$$H(x_i)=f(x_i),H'(x_i)=f'(x_i),(i=0,1,\cdots,n)$$

则有

$$f(x)-H(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}[\pi(x)]^2$$

其中 $\xi\in[a,b]$，由于高斯求积公式 $\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i f_i$ 具有 $2n+1$ 次代数精度，所以它对 $2n+1$ 次埃尔米特插值多项式 $H(x)$能精确成立，即

$$\int_a^b H(x)\mathrm{d}x=\sum_{i=0}^{n}\omega_i H(x_i)$$

又由插值条件

$$H(x_i)=f(x_i)$$

有

$$\int_a^b H(x)\mathrm{d}x=\sum_{i=0}^{n}\omega_i f(x_i)$$

则高斯求积公式的余项为

$$
\begin{aligned}
R[f] &= \int_a^b f(x)\mathrm{d}x - \sum_{i=0}^{n} \omega_i f_i \\
&= \int_a^b f(x)\mathrm{d}x - \int_a^b H(x)\mathrm{d}x \\
&= \int_a^b [f(x) - H(x)]\mathrm{d}x \\
&= \int_a^b \frac{f^{(2n+2)}(\xi)}{(2n+2)!}[\pi(x)]^2 \mathrm{d}x
\end{aligned}
$$

因为 $f^{(2n+2)}(x)$在$[a,b]$上连续,$[\pi(x)]^2 \geqslant 0$ 在$[a,b]$上可积且不变号,由积分中值定理,必有 $\eta \in [a,b]$使

$$
R[f] = \frac{f^{(2n+2)}(\eta)}{(2n+2)!}\int_a^b [\pi(x)]^2 \mathrm{d}x
$$

证毕.

三、高斯求积公式的稳定性与收敛性

由于高斯点往往是无理数,在计算 $f(x_i)$时常有舍入误差,故人们非常关心高斯求积公式的稳定性.事实上,与 N-C 公式相比,高斯求积公式不但具有高精度,而且是数值稳定的求积公式.高斯求积公式的稳定性之所以能够得到保证,是由于它的求积系数具有非负性.

定理 3.10 高斯求积公式中的求积系数 $\omega_i (i=0,1,\cdots,n)$全是正的,且有

$$
\omega_i = \int_a^b [l_i(x)]^2 \mathrm{d}x, (i=0,1,\cdots,n)
$$

其中 $l_i(x)$是 n 次拉格朗日插值基函数.

证明 因为 $l_i(x)$是 n 次多项式,所以 $l_i^2(x)$是 $2n$ 次多项式,

而高斯求积公式(3.1)具有 $2n+1$ 次代数精度,故对 $l_i^2(x)$ 能精确成立,即有

$$\int_a^b l_i^2(x)\mathrm{d}x=\sum_{k=0}^{n}\omega_k l_i^2(x_k) \tag{3.46}$$

注意到 $l_i(x_k)=\delta_{ik}$,则式(3.46)右端等于 ω_i,从而有

$$\omega_i=\int_a^b l_i^2(x)\mathrm{d}x>0$$

即求积系数 ω_i 全为正的.

因为高斯求积公式是插值型求积公式,则由定理 3.5 得

推论 3.2 高斯求积公式是稳定的.

关于收敛性,有结论:若 $f(x)$ 在区间 $[a,b]$ 上连续,则高斯求积公式收敛,且收敛到 $\int_a^b f(x)\mathrm{d}x$,即

$$\lim_{n\to\infty}\sum_{i=0}^{n}\omega_i f(x_i)=\int_a^b f(x)\mathrm{d}x$$

四、带权的高斯求积公式

为了得到更一般的高精度求积公式,考虑带权的积分

$$I[f]=\int_a^b \rho(x)f(x)\mathrm{d}x \tag{3.47}$$

其中 $\rho(x)\geqslant 0$ 是 $[a,b]$ 上的权函数,当 $\rho(x)=1$ 时,也称为不带权的积分.

若求积公式

$$\int_a^b \rho(x)f(x)\mathrm{d}x\approx\sum_{i=0}^{n}\omega_i f(x_i) \tag{3.48}$$

具有最高代数精度 $2n+1$ 次,则称它为 $[a,b]$ 上带权 $\rho(x)$ 的高斯求积公式,其求积节点 x_i $(i=0,1,\cdots,n)$ 仍称为高斯点,ω_i $(i=0,1,\cdots,n)$ 称为高斯求积系数.

可以证明，前面讨论的不带权的高斯公式的一些结论对于带权的高斯积分也基本适用，只需要在前面讨论的积分号下加一个权函数 $\rho(x)$，并把"正交"改成"对权函数 $\rho(x)$ 正交"即可．例如，也有类似于定理 3.8 的结论，但这些高斯点是带权 $\rho(x)$ 的 $n+1$ 次正交多项式的零点．根据正交多项式的理论，这些零点是 $n+1$ 个单实零点，且全都在区间 $[a,b]$ 内．

例如，当 $[a,b]=[-1,1]$，$\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$ 时的正交多项式为切比雪夫正交多项式 $T_n(x)=\cos(n\arccos x)$，此时的高斯型求积公式称为高斯-切比雪夫求积公式，可表示为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\frac{\pi}{n+1}\sum_{i=0}^{n}f(x_i)$$

其中
$$x_i=\cos\left(\frac{2i+1}{2(n+1)}\pi\right),(i=0,1,\cdots,n)$$

为 $T_{n+1}(x)$ 的零点，这个公式由于求积系数 $\omega_i=\dfrac{\pi}{n+1}$ 计算简单，又因被积函数包含因子 $(1-x^2)^{-\frac{1}{2}}$，故可处理此类因子的奇异积分．

例 3.7 求形如 $\displaystyle\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\omega_0 f(x_0)+\omega_1 f(x_1)$ 的两点高斯型求积公式．

解 显然，所求的高斯点是 $[-1,1]$ 上带权 $\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$ 的二次切比雪夫正交多项式 $T_2(x)=\cos(2\arccos x)=2x^2-1$ 的零点

$$x_0=-\frac{\sqrt{2}}{2},\ x_1=\frac{\sqrt{2}}{2}$$

则所求的高斯型求积公式为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\omega_0 f\left(-\frac{\sqrt{2}}{2}\right)+\omega_1 f\left(\frac{\sqrt{2}}{2}\right)$$

为了求出 ω_0,ω_1，令上式对 $f(x)=1,x$ 精确成立，可得 $\omega_0=\omega_1=\frac{\pi}{2}$，则所求的高斯型求积公式为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\frac{\pi}{2}\left[f\left(-\frac{\sqrt{2}}{2}\right)+f\left(\frac{\sqrt{2}}{2}\right)\right].$$

高斯求积法的优点是积分精度高，精度不够时也可以复合，计算数值稳定，当被积函数 $f(x)$ 连续时积分总是收敛的，且收敛速度较快. 高斯积分还可以处理一些近似于奇异积分的问题. 但高斯积分的节点与系数的计算比较麻烦 .

3.6　数值微分

由函数在一些离散点上的函数值来推算出函数在某点处的导数近似值，这类问题称为数值微分 .

当函数 $f(x)$ 用表格形式给出时，通常只能用近似方法求其数值微分. 下面介绍两种基本的（也是常用的）数值微分方法 .

一、用插值多项式求数值微分

构造数值微分公式的一种常用方法是用一个易于计算其微分的函数近似代替原问题中的函数 $f(x)$. 显然，用插值多项式来构造数值微分公式是比较合适的 .

设已知函数 $f(x)$ 在 $n+1$ 个互异节点 x_i 上取值为 $f(x_i)=y_i$，$(i=0,1,\cdots,n)$，应用插值法，可以构造插值多项式 $P_n(x)$ 满足条件 $P_n(x_i)=y_i$，则有

$$f(x)=P_n(x)+R_n(x)=P_n(x)+\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)$$

从而

$$f'(x)=P_n'(x)+R_n'(x)$$

由于多项式的求导比较容易,我们可取 $P_n'(x)$ 的值作为 $f'(x)$ 的近似值,这样建立的数值微分公式

$$f'(x)\approx P_n'(x) \tag{3.49}$$

就称为插值型的数值微分公式.

数值微分公式(3.49)的余项为

$$\begin{aligned}R_n'(x)=f'(x)-P_n'(x)&=\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\right]\\&=\frac{f^{(n+1)}(\xi)}{(n+1)!}\frac{\mathrm{d}}{\mathrm{d}x}\prod_{j=0}^{n}(x-x_i)\\&\quad+\frac{1}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\frac{\mathrm{d}}{\mathrm{d}x}f^{(n+1)}(\xi)\end{aligned} \tag{3.50}$$

在这一余项公式中,由于 ξ 是 x 的未知函数,因此对任意的 x,我们无法求出它的第二项

$$\frac{1}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\frac{\mathrm{d}}{\mathrm{d}x}f^{(n+1)}(\xi)$$

的数值,从而也无法估计误差 $R_n'(x)$. 但是,如果限定为求某个节点 $x_k(k=0,1,2,\cdots,n)$ 上的导数值时,那么式(3.50)的第二项必然等于零,这时有余项公式

$$\begin{aligned}R_n'(x_k)&=f'(x_k)-P_n'(x_k)\\&=\frac{f^{(n+1)}(\xi)}{(n+1)!}\left[\frac{\mathrm{d}}{\mathrm{d}x}\prod_{j=0}^{n}(x-x_j)\right]_{x=x_k}\\&=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{\substack{j=0\\j\neq k}}^{n}(x_k-x_j)\end{aligned} \tag{3.51}$$

从而可建立一系列在节点 x_k 处的数值微分公式：

$$f'(x_k)=P_n'(x_k)+R_n'(x_k),\quad (k=0,1,2,\cdots,n)$$

用插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似函数，还可以建立高阶的数值微分公式：

$$f^{(k)}(x)\approx P_n^{(k)}(x),\ (k=1,2,\cdots) \tag{3.52}$$

为简化讨论，一般只考虑节点处的导数值，且通常取节点是等距的，即

$$x_k=x_0+kh,\quad (k=0,1,2,\cdots,n)$$

下面给出用插值多项式导出的几个常用数值微分公式．

1．两点公式

如果 $f'(x)$ 连续，$f''(x)$ 存在，且已知两个节点 x_0,x_1 上的函数值 $f(x_0)$，$f(x_1)$，则可作线性插值多项式

$$P_1(x)=\frac{x-x_1}{x_0-x_1}f(x_0)+\frac{x-x_0}{x_1-x_0}f(x_1)$$

来代替函数 $f(x)$.

对上式两端求导，记

$$x_1=x_0+h,$$

于是就有近似数值微分公式：

$$f'(x)\approx P_1'(x)=\frac{1}{h}[f(x_1)-f(x_0)]$$

余项是

$$R_1'(x_k)=f'(x_k)-P_1'(x_k)$$

$$=(2x_k-x_0-x_1)\frac{f''(\xi)}{2!},\ (k=0,1)$$

因此得到带余项的两点数值微分公式：

$$\begin{cases} f'(x_0)=\dfrac{1}{h}[f(x_1)-f(x_0)]-\dfrac{h}{2}f''(\xi) \\ f'(x_1)=\dfrac{1}{h}[f(x_1)-f(x_0)]+\dfrac{h}{2}f''(\xi), \quad (x_0<\xi<x_1) \end{cases} \tag{3.53}$$

2. 三点公式

如果 $f''(x)$ 连续，$f'''(x)$ 存在，且在三个节点 $x_k=x_0+kh$，$(k=0,1,2)$ 上函数值 $f(x_k)$ 已知，那么可作二次插值多项式 $P_2(x)$，用 $P_2(x)\approx f(x)$，再对两端求导，有

$$f'(x)\approx P_2'(x)=f(x_0)\frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)}$$

$$+f(x_1)\frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)}$$

$$+f(x_2)\frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}$$

$$R_2(x_k)=\frac{f'''(\xi)}{3!}[(x_k-x_0)(x_k-x_1)+(x_k-x_0)(x_k-x_2)$$

$$+(x_k-x_1)(x_k-x_2)], \ \xi\in[x_0,x_2]$$

分别取 $x=x_0,x_1$ 和 x_2，即得带余项的三点数值微分公式：

$$\begin{cases} f'(x_0)=\dfrac{1}{2h}[-3f(x_0)+4f(x_1)-f(x_2)]+\dfrac{h^2}{3}f'''(\xi) \\ f'(x_1)=\dfrac{1}{2h}[-f(x_0)+f(x_2)]-\dfrac{h^2}{6}f'''(\xi) \\ f'(x_2)=\dfrac{1}{2h}[f(x_0)-4f(x_1)+3f(x_2)]+\dfrac{h^2}{3}f'''(\xi) \end{cases}$$

$$\xi\in(x_0, x_2) \tag{3.54}$$

类似地，可推导出实用的五点数值微分公式．

但应注意：当插值多项式 $P_n(x)$ 收敛于 $f(x)$ 时，不能保证 $P'_n(x)$ 一定收敛于 $f'(x)$，而且当节点间的距离缩小时，虽然截断误差缩小了，但舍入误差却可能增大．因此，缩小步长不一定都能提高计算结果的精确度．

例 3.8 已知函数 $f(x)=e^x$ 的下列数值

x	0	0.90	0.99	1.00	1.01	1.10	2
$f(x)=e^x$	1.000	2.460	2.691	2.718	2.746	3.004	7.389

试用三点数值微分公式(3.54)中的第二式计算 $f'(1)$ 的近似值并比较步长 h 分别取 1，0.1，0.01 的计算结果．

解 (1) $h=1$ 时，取 $x_0=0$，$x_1=1$，$x_2=2$

$$f'(1)\approx\frac{1}{2}(-e^0+e^2)=3.195$$

(2) $h=0.1$ 时，取 $x_0=0.90$，$x_1=1.00$，$x_2=1.10$

$$f'(1)\approx\frac{1}{2\times0.1}(-e^{0.90}+e^{1.10})=2.720$$

(3) $h=0.01$ 时，取 $x_0=0.99$，$x_1=1.00$，$x_2=1.01$

$$f'(1)\approx\frac{1}{2\times0.01}(-e^{0.99}+e^{1.01})=2.750$$

容易求得 $f'(1)$ 的真值为 $e^1=2.7182818$.

上面的计算结果表明，当步长由 1 减少到 0.1 时，计算精度明显提高，但是当步长由 0.1 减少到 0.01 时，精度反而有所降低．问题的根源在于在实际计算中，不但有截断误差，还有舍入误差，而数值微分恰好对舍入误差非常敏感，它随 h 的缩小而增大，这就是造成计算不稳定性的原因．故使用数值微分公式时，要特别注意误差的分析．

二、用样条函数求数值微分

上面介绍的数值微分方法不便于计算节点之间点处的导数值，而由第二章 2.5 节的讨论知，当步长 $h \to 0$ 时，三次样条插值函数 $S(x)$ 及其一、二阶导数均一致收敛于被插值函数 $f(x)$ 及其一、二阶导数，故用样条函数的导数近似函数导数：

$$f^{(k)}(x) \approx S^{(k)}(x), \quad (k=1,2,\cdots)$$

不仅可靠性好，而且可以计算非节点处导数的近似值．可以证明其截断误差为

$$f^{(k)}(x) - S^{(k)}(x) = O(h^{4-k})$$

用三次样条插值函数求数值微分其过程如下：

从给定的函数表

x	x_0	x_1	$\cdots$	x_n
$y=f(x)$	y_0	y_1	$\cdots$	y_n

和适当的边界条件出发，按第二章 2.5 节样条函数插值中介绍的方法，构造一个三次样条插值函数 $S(x)$，近似代替 $f(x)$，由于 $S(x)$ 是一个分段三次多项式，在各子区间上容易求导，所以可直接通过 $S(x)$ 在每一个小区间上的表达式 $S_i(x)$ 求导得 $S_i'(x)$，用 $S_i'(x)$ 近似代替 $f'(x)$，则

$$S'(x) \approx f'(x)$$

即为相应数值微分公式，若只需求节点的导数，则

$$f'(x_i) \approx S'(x_i)$$

数值微分公式的重要应用，是求微分方程的数值解．

学习指导

一、基本要求与重点

1. 掌握数值积分和数值微分的基本思想、基本方法和基本理论。

2. 掌握衡量求积公式精确度的标准——“代数精确度”的概念,并会熟炼的确定数值求积公式的代数精确度.

3. 熟炼掌握等距节点下的求积公式:牛顿-柯特斯公式(包括复合求积公式)、龙贝格公式和不等距节点下的求积公式——高斯求积公式,并要求会运用它们计算定积分的数值解并估计误差.

4. 龙贝格求积公式是李查逊外推算法的一个具体应用,应从龙贝格求积算法中体会数值方法的加速收敛技巧的基本思想.

5. 掌握数值微分公式的导出及几个常用的数值微分公式,在使用各种数值微分公式计算时,要特别注意误差分析.

6. 在实际使用数值积分公式时,能注重对计算结果的准确性、稳定性、收敛性、工作量等因素的全面综合考虑,要求学会能根据不同的具体问题去选择合适的数值积分公式.

本章重点:定积分的数值计算,代数精确度,数值微分.

二、例题分析与解答

例 1 对定积分 $\int_0^3 f(x)\mathrm{d}x$ 试构造一个至少具有三次代数精确度的数值求积公式.

解 已知含有 $n+1$ 个节点的插值型求积公式至少具有 n 次代数精确度.按题设要求,若在区间$[0,3]$上取 4 个节点,构造其插值型求积公式为

$$\int_0^3 f(x)\mathrm{d}x \approx \omega_0 f(x_0) + \omega_1 f(x_1) + \omega_2 f(x_2) + \omega_3 f(x_3) \tag{1}$$

则其至少有 3 次代数精确度. 为简单起见,不妨取 $x_0=0, x_1=1, x_2=2, x_3=3$.

则 $$\omega_0 = \int_0^3 l_0(x)\mathrm{d}x = \int_0^3 \frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)}\mathrm{d}x = \frac{3}{8},$$

$$\omega_1 = \int_0^3 l_1(x)\mathrm{d}x = \int_0^3 \frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)}\mathrm{d}x = \frac{9}{8},$$

$$\omega_2 = \int_0^3 l_2(x)\mathrm{d}x = \int_0^3 \frac{(x-0)(x-2)(x-3)}{(2-0)(2-1)(2-3)}\mathrm{d}x = \frac{9}{8},$$

$$\omega_3 = \int_0^3 l_3(x)\mathrm{d}x = \int_0^3 \frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)}\mathrm{d}x = \frac{3}{8}.$$

将 $x_i, \omega_i (i=0,1,2,3)$ 代入式(1),得求积公式为

$$\int_0^3 f(x)\mathrm{d}x \approx \frac{3}{8}f(0) + \frac{9}{8}f(1) + \frac{9}{8}f(2) + \frac{3}{8}f(3) \tag{2}$$

式(2)具有至少三次的代数精度,为进一步确定它的精度,可令 $f(x)=x^4$ 代入,显然式(2)不能精确成立,所以求积公式(2)具有三次代数精度.

当然,此题也可直接由代数精度的定义:依次取 $f(x)=1, x, x^2, x^3$ 代入式(1),并令 $x_i=i (i=0,1,2,3)$,则得关于求积系数 ω_i $(i=0,1,2,3)$ 的线性代数方程组:

$$\begin{cases} \omega_0 + \omega_1 + \omega_2 + \omega_3 = \int_0^3 1 \cdot \mathrm{d}x = 3 \\ \omega_1 + 2\omega_2 + 3\omega_3 = \int_0^3 x\mathrm{d}x = \frac{9}{2} \\ \omega_1 + 4\omega_2 + 9\omega_3 = \int_0^3 x^2\mathrm{d}x = 9 \\ \omega_1 + 8\omega_2 + 27\omega_3 = \int_0^3 x^4\mathrm{d}x = \frac{81}{4} \end{cases}$$

解此方程组得

$$\omega_0=\frac{3}{8},\omega_1=\frac{9}{8},\omega_2=\frac{9}{8},\omega_3=\frac{3}{8},$$

代入式(1)，同样得到具有三次代数精确度的求积公式(2)。

例 2 计算积分

$$I=\int_1^3(x^3-2x^2+7x-5)\mathrm{d}x.$$

解 此题可用牛顿-莱布尼兹公式计算出 I 的精确值：

$$I=\left[\frac{x^4}{4}-\frac{2x^3}{3}+\frac{7x^2}{2}-5x\right]_1^3=20\frac{2}{3}.$$

下面用几个低阶的牛顿-柯斯特求积公式计算其积分值：

取 $n=1$，即梯形公式(3.6)，有表(例)3-1.

表(例)3-1

x	1	3
$f(x)$	1	25

则 $T=\frac{3-1}{2}\times(1+25)=26.$

取 $n=2$，即抛物线求积公式(3.7)，有表(例)3-2.

表(例)3-2

x	1	2	3
$f(x)$	1	9	25

则 $S=\frac{3-1}{6}(1+4\cdot 9+25)=\frac{62}{3}=20\frac{2}{3}.$

取 $n=4$，即柯特斯公式(3.8)，有表(例)3-3.

表(例)3-3

x	1	$\frac{3}{2}$	2	$\frac{5}{2}$	3
$f(x)$	1	$\frac{35}{8}$	9	$\frac{125}{8}$	9

则
$$C=\frac{3-1}{90}\times\left(7\times 1+32\times\frac{35}{8}+12\times 9+32\times\frac{125}{8}+7\times 9\right)=20\ \frac{2}{3}.$$

可以看出,虽然牛顿-柯斯特公式是数值求积公式,但是在这个例子中,对于同一个积分,当 $n\geqslant 2$ 时,公式却是精确的,这是由于抛物线公式具有三次代数精度,柯特斯公式具有五次代数精度,它们对被积函数为三次多项式当然是精确成立的.

例 3 计算积分 $\int_0^{\pi}\sin x\mathrm{d}x$,若用复合抛物线求积公式,问积分区间要多少等分才能保证误差不超过 2×10^{-5}?

解 从误差估计式(3.16)知

$$R_s=\int_a^b f(x)\mathrm{d}x-S_n=-\frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\eta),\eta\in(a,b)$$

由于 $f(x)=\sin x, f^{(4)}(x)=\sin x, h=\frac{\pi}{n}, b-a=\pi$,当 $x\in[0,\pi]$ 时,有 $|f^{(4)}(x)|=|\sin x|\leqslant 1$,所以

$$\begin{aligned}|R_s[f]|&=\left|-\frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\eta)\right|\\&=\left|-\frac{b-a}{180}\left(\frac{h}{2}\right)^4\sin\eta\right|\leqslant\frac{\pi}{180}\left(\frac{\pi}{2n}\right)^4\\&=\frac{\pi^5}{2\,880}\times\frac{1}{n^4}\end{aligned}$$

若要误差不超过 2×10^{-5}，只要

$$n^4 \geqslant \frac{\pi^5}{2\,880\times2\times10^{-5}}$$

可解出 $n\geqslant9$，为了计算简便，可取 $n=10$，则 $h=\frac{\pi}{10}$，即将区间$[0,\pi]$10 等分，就能保证误差满足给定的精度要求.

事实上，按复合抛物线求积公式(3.15)计算，有

$$\int_0^{\pi}\sin x\mathrm{d}x \approx \frac{\pi}{60}\left[2\sum_{i=1}^{9}\sin\frac{i\pi}{10}+4\sum_{i=1}^{10}\sin\frac{(2i-1)\pi}{20}\right]$$

$$=2.000\,007.$$

而本题精确解为 $\int_0^{\pi}\sin x\mathrm{d}x=[-\cos x]_0^{\pi}=2$，可见上述计算已达到了提出的误差要求.

如果误差要求相同，而按复合梯形公式(3.13)计算，则由其误差估计式(3.14)得

$$|R_T[f]|=\left|-\frac{b-a}{12}h^2f''(\eta)\right|=\left|-\frac{\pi h^2}{12}(-\sin\eta)\right|$$

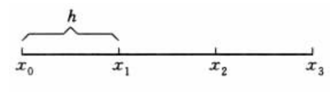
$$\leqslant\frac{\pi^8}{12n^2}\leqslant2\times10^{-5}$$

即 $n\geqslant\frac{\pi^3}{24}\times10^5$，可解得 $n\geqslant360$，可见其计算量大大超过复合抛物线公式，这说明复合抛物线求积公式的精度较高，故在实际计算时，较多地应用复合抛物线公式.

例 4 用 x_n 表示 $a+nh$，f_n 表示 $f(x_n)$，试推导一个开型的牛顿-柯斯特求积公式：

$$\int_{x_0}^{x_3}f(x)\mathrm{d}x\approx\frac{3}{2}h(f_1+f_2)$$

解 在求积公式 $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n}\omega_i f(x_i)$ 中，节点 $x_i\in[a,b](i=0,1,2,\cdots,n)$，若在区间$[a,b]$的两个端点，即 $x_0=a,x_n=b$ 处要计算函数 $f(x)$的值 $f(x_0)$与 $f(x_n)$，则求积公式称为闭型求积公式，例如，牛顿-柯斯特公式(3.5)为闭型公式. 若不在积分区间$[a,b]$的两个端点处计算函数值，那么，求积公式(3.1)称为开型求积公式. 本题要推导求积节点和求积系数都已知的开型 N-C 求积公式，可用插值法：



将$[x_0,x_3]$分为三等分，$x_i=x_0+ih,(i=0,2,3),h=\dfrac{x_3-x_0}{3}$，取 x_1,x_2 为插值节点，作被积函数 $f(x)$的一次插值，如取 $f(x)\approx L_1(x)$，$L_1(x)=l_1(x)f(x_1)+l_2(x)f(x_2)=\dfrac{x-x_2}{x_1-x_2}f(x_1)+\dfrac{x-x_1}{x_2-x_1}f(x_2)$，则 $\int_{x_0}^{x_3}f(x)\mathrm{d}x\approx\int_{x_0}^{x_3}L_1(x)\mathrm{d}x=\omega_1 f_1+\omega_2 f_2$.

又，$x_1=x_0+h,x_2=x_0+2h$，设 $x=x_0+sh,(0\leqslant s\leqslant 3)$

所以
$$\omega_1=\int_{x_0}^{x_3}l_1(x)\mathrm{d}x=\int_{x_0}^{x_3}\frac{x-x_2}{x_1-x_2}\mathrm{d}x=\int_0^3\frac{(s-2)h}{-h}h\,\mathrm{d}s$$
$$=-h\int_0^3(s-2)\mathrm{d}s=-h\left.\frac{(s-2)^2}{2}\right|_0^3=\frac{3}{2}h$$
$$\omega_2=\int_{x_0}^{x_3}l_2(x)\mathrm{d}x=\int_{x_0}^{x_3}\frac{x-x_1}{x_2-x_1}\mathrm{d}x=\int_0^3\frac{(s-1)h}{h}h\,\mathrm{d}s$$
$$=\frac{3}{2}h$$

从而得 $\int_{x_0}^{x_3}f(x)\mathrm{d}x\approx\dfrac{3}{2}h(f_1+f_2)$.

若对本题中的区间$[x_0, x_3]$三等分，分别用闭型的 N-C 公式和开型的 N-C 公式计算积分 $I=\int_0^{\frac{\pi}{4}} \sin x \mathrm{d}x = 1-\frac{\sqrt{2}}{2}$的近似值，可得闭型 N-C 公式的计算值为

$I \approx 0.29291070$，误差为 0.00001718.

而开型 N-C 公式的计算值为

$I \approx 0.29286923$，误差为 0.00002399.

一般来说，闭型 N-C 求积公式的计算结果比开型 N-C 求积公式的计算结果好. 因此，在实际应用中大多采用闭型 N-C 求积公式，但开型 N-C 求积公式在数值求解常微分方程中很有用.

例 5 设 $f^{(4)}(x)$在$[a,b]$上连续，试利用埃尔米特插值公式导出带有导数值的求积公式：

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a)+f(b)] + \frac{(b-a)^2}{12}[f'(a)-f'(b)]$$

及其误差表示式：

$$R[f]=\frac{(b-a)^5}{4!30}f^{(4)}(\eta), (a \leqslant \eta \leqslant b)$$

解 取 $x_0=a, x_1=b$ 为插值节点，作 $f(x)$的三次埃尔米特插值多项式 $H_3(x)$，使满足条件：

$$H_3(x_0)=f(x_0)=f(a), \quad H_3(x_1)=f(x_1)=f(b),$$

$$H_3'(x_0)=f'(x_0)=f'(a), \quad H_3'(x_1)=f'(x_1)=f'(b)$$

则由公式(2.24)得

$$H_3(x)=\sum_{k=0}^{1}[1-2l_k'(x_k)(x-x_k)]l_k^2(x)f_k + \sum_{k=0}^{1}(x-x_k)l_k^2(x)f_k'$$

$$=\left[1+2\left(\frac{x-a}{b-a}\right)\right]\left(\frac{x-b}{a-b}\right)^2 f(a)$$

$$+\left[1+2\left(\frac{x-b}{a-b}\right)\right]\left(\frac{x-a}{b-a}\right)^2 f(b)$$

$$+\frac{(x-a)(x-b)^2}{(a-b)^2}f'(a)$$

$$+\frac{(x-b)(x-a)^2}{(b-a)^2}f'(b)$$

所以

$$\int_a^b f(x)\mathrm{d}x \approx \int_a^b H_3(x)\mathrm{d}x=\frac{f(a)}{(a-b)^2}\int_a^b (x-b)^2\mathrm{d}x$$

$$+\frac{f(a)}{(b-a)^3}\int_a^b (x-a)(x-b)^2\mathrm{d}x$$

$$+\frac{f(b)}{(a-b)^2}\int_a^b (x-a)^2\mathrm{d}x$$

$$+\frac{2f(b)}{(a-b)^3}\int_a^b (x-b)(x-a)^2\mathrm{d}x$$

$$+\frac{f'(a)}{(a-b)^2}\int_a^b (x-a)(x-b)^2\mathrm{d}x$$

$$+\frac{f'(b)}{(a-b)^2}\int_a^b (x-b)(x-a)^2\mathrm{d}x$$

$$=\frac{b-a}{2}[f(a)+f(b)]+\frac{(b-a)^2}{12}[f'(a)-f'(b)]$$

再证误差

由埃尔米特插值的误差公式(2.27)**得**

$$R_3(x)=f(x)-H_3(x)=\frac{1}{4!}f^{(4)}(\xi)(x-a)^2(x-b)^2$$

所以 $R[f]=\int_a^b R_3(x)\mathrm{d}x=\int_a^b \frac{1}{4!}f^{(4)}(\xi)(x-a)^2(x-b)^2\mathrm{d}x.$

因为 $f^{(4)}(\xi)$ 在 $[a,b]$ 上连续，$(x-a)^2(x-b)^2$ 在 $[a,b]$ 上不变号，由积分中值定理，存在 $\eta\in(a,b)$，使

$$R[f]=\frac{1}{4!}f^{(4)}(\eta)\int_a^b (x-a)^2(x-b)^2\mathrm{d}x$$

$$=\frac{(b-a)^5}{4!\,30}f^{(4)}(\eta).$$

例 6　用龙贝格算法求积计算积分

$$I=\int_0^1 \frac{4}{1+x^2}\mathrm{d}x.$$

解　$f(x)=\frac{4}{1+x^2}, a=0, b=1$

(1) $T_1=\frac{1}{2}[f(0)+f(1)]=\frac{1}{2}(4+2)=3$

(2) 由 $f\left(\frac{1}{2}\right)=\frac{16}{5}$，则

$$T_2=\frac{1}{2}T_1+\frac{1}{2}f\left(\frac{1}{2}\right)=\frac{1}{2}\left(3+\frac{16}{5}\right)=3.1$$

(3) $S_1=\frac{4}{3}T_2-\frac{1}{3}T_1=3.13333$

再计算　$T_4=\frac{1}{2}T_2+\frac{1}{4}\left[f\left(\frac{1}{4}\right)+f\left(\frac{3}{4}\right)\right]=3.13118$

由此得　$S_2=\frac{4}{3}T_4-\frac{1}{3}T_2=3.14157$

$$C_1=\frac{16}{15}S_2-\frac{1}{15}S_1=3.14212$$

再计算 $T_8=\frac{1}{2}T_4+\frac{1}{8}\left[f\left(\frac{1}{8}\right)+f\left(\frac{3}{8}\right)+f(\frac{5}{8})\right.$

$$\left.+f\left(\frac{7}{8}\right)\right]=3.13899$$

从而算得 $S_4=\frac{4}{3}T_8-\frac{1}{3}T_4=3.14159$

$$C_2=\frac{16}{15}S_4-\frac{1}{15}S_2=3.14159$$

$$R_1=\frac{64}{63}C_2-\frac{1}{63}C_1=3.14158$$

(4) 把区间再分半,重复上面的计算步骤,可得 $T_{16}=3.14094$, $S_8=3.14159$,$R_2=3.14159$.

由 $|R_2-R_1|\leqslant 0.00001$,已精确到小数点后五位,故可取 $I=\int_0^1\frac{4}{1+x^2}dx\approx 3.14159$.

这一结果与 I 的准确值 $\int_0^1\frac{4}{1+x^2}dx=4\arctan x\Big|_0^1=\pi$ 相比较已有较好的精度.

上述计算结果见表(例)3-4.

表(例)3-4　　数值表

k	区间个数 2^k	T_{2^k}	$S_{2^{k-1}}$	$C_{2^{k-2}}$	$R_{2^{k-3}}$
0	1	3.00000			
1	2	3.10000	3.13333		
2	4	3.13118	3.14157	3.14212	
3	8	3.13899	3.14159	3.14159	3.14158
4	16	3.14094	3.14159	3.14159	3.14159

例 7　用高斯-勒让德求积公式(节点数 $n=2,3$)计算定积分

$$I[f]=\int_0^1 x^2 e^x dx.$$

解　高斯-勒让德公式的高斯点 x_i 和求积系数 ω_i 都有表可查，例如表(例)3-5 给出了 1～3 个节点时高斯-勒让德公式的节点 x_i 和系数 ω_i.

表(例)3-5　　　　求积节点和系数表

节点数 n	节点 x_i	系数 ω_i
1	0.0000000	2.0000000
2	±0.5773503	1.0000000
3	±0.7745967 0.0000000	0.5555556 0.8888889

本题所给积分区间为[0,1]**，所以，必须先作变量替换** $x=\frac{1}{2}(1+t)$**，于是**

$$I(f)=\int_0^1 x^2 e^x dx=\frac{1}{2}\int_{-1}^1\left[\frac{1}{2}(t+1)\right]^2 e^{\frac{1}{2}(1+t)}dt,$$

当 $n=2$ **时，用表(例)3-5 中的高斯点和求积系数，可得**

$$\begin{aligned}I[f]&\approx\frac{1}{2}\sum_{i=0}^{1}\omega_i f(x_i)\\&=\frac{1}{8}\left[(1+0.5773503)^2 e^{\frac{1}{2}(1+0.5773503)}\right.\\&\quad\left.+(1-0.5773503)^2 e^{\frac{1}{2}(1-0.5773503)}\right]\\&=0.7119418\end{aligned}$$

当 $n=3$ **时，用表(例)3-5 中的高斯点和求积系数，可得**

$$I[f]\approx\frac{1}{2}\sum_{i=0}^{2}\omega_i f(x_i)$$

$$=\frac{1}{8}[(0.5555556)\times(1+0.7745967)^2 e^{\frac{1}{2}(1+0.7745967)}$$

$$+0.8888889\times(1+0.0000000)^2 e^{\frac{1}{2}(1+0.0000000)}$$

$$+0.5555556\times(1-0.7745967)^2 e^{\frac{1}{2}(1-0.7745967)}]$$

$$\approx 0.7182518$$

容易计算出定积分的精确值：

$$I[f]=e-2\approx 0.7182818.$$

例 8 （1）**证明切比雪夫**（Chebyshev）**正交多项式** $T_n(x)=\cos(n\arccos x)$**具有三项递推关系式：**

$$\begin{cases} T_0(x)=1, T_1(x)=x \\ T_{n+1}(x)=2xT_n(x)-T_{n-1}(x), (n=1,2,\cdots) \end{cases} \tag{3}$$

（2）**求形如** $\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}dx\approx\omega_0 f(x_0)+\omega_1 f(x_1)+\omega_2 f(x_2)$**的三点高斯-切比雪夫求积公式.**

（3）**试用** n **点高斯-切比雪夫求积公式计算积分**

$$I[f]=\int_0^2\frac{x^2-x+1}{\sqrt{x(2-x)}}dx,$$

问当节点数 n **取何值时，能得到积分的精确值？并计算之.**

解 （1）**由** $T_n(x)=\cos(n\arccos x)$，**显然有当** $n=0$ **时，**$T_0(x)=1$；$n=1$ **时，**$T_1(x)=x$.

当 $n\geqslant 1$ **时，令** $x=\cos\theta$，**则** $T_n(x)=\cos n\theta$，**由三角恒等式** $\cos(n+1)\theta+\cos(n-1)\theta=2\cos\theta\cos n\theta$，**即** $T_{n+1}(x)+T_{n-1}(x)=2xT_n(x)$，**移项就得递推关系式**(3).

（2）**切比雪夫多项式** $T_n(x)=\cos(n\arccos x)$**是区间**$[-1,1]$**上关于权函数** $\rho(x)=\frac{1}{\sqrt{1-x^2}}$**的** n **次正交多项式，以它的** n **个零点**

作为求积节点得到的插值型求积公式称为高斯-切比雪夫求积公式. 显然,本题所求的高斯点是三次切比雪夫正交多项式 $T_3(x)=\cos(3\arccos x)=4x^3-3x$ 的零点:

$$x_0=-\frac{\sqrt{3}}{2}, x_1=0, x_2=\frac{\sqrt{3}}{2}$$

则所求的高斯型求积公式为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\omega_0 f\left(-\frac{\sqrt{3}}{2}\right)+\omega_1 f(0)+\omega_2 f\left(\frac{\sqrt{3}}{2}\right).$$

为了求出 $\omega_0,\omega_1,\omega_2$(因为该求积公式具有五次代数精度),可令上式对 $f(x)=1,x,x^2$ 精确成立,即

$$\begin{cases} f(x)=1 \text{ 时,有 } \omega_0+\omega_1+\omega_2=\int_{-1}^{1}\frac{\mathrm{d}x}{\sqrt{1-x^2}}=\pi, \\ f(x)=x \text{ 时,有 } -\frac{\sqrt{3}}{2}\omega_0+0+\frac{\sqrt{3}}{2}\omega_2 \\ \qquad =\int_{-1}^{1}\frac{x}{\sqrt{1-x^2}}\mathrm{d}x=0, \\ f(x)=x^2 \text{ 时,有 } \frac{3}{4}\omega_0+0+\frac{3}{4}\omega_2 \\ \qquad =\int_{-1}^{1}\frac{x^2}{\sqrt{1-x^2}}\mathrm{d}x=\frac{\pi}{2}. \end{cases}$$

从而可解得 $\omega_0=\omega_1=\omega_2=\frac{\pi}{3}$.

所以,三点的高斯-切比雪夫求积公式为

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x\approx\frac{\pi}{3}\left[f\left(-\frac{\sqrt{3}}{2}\right)+f(0)+f\left(\frac{\sqrt{3}}{2}\right)\right].$$

(3) 先作变换 $x=\frac{a+b}{2}+\frac{b-a}{2}t=1+t$ 把积分区间[0,2]化为

区间$[-1,1]$，则

$$I[f]=\int_0^2\frac{x^2-x+1}{\sqrt{x(2-x)}}\mathrm{d}x$$

$$=\int_{-1}^1\frac{(1+t)^2-(1+t)+1}{\sqrt{(1+t)[2-(1+t)]}}\mathrm{d}t$$

$$=\int_{-1}^1\frac{t^2+t+1}{\sqrt{1-t^2}}\mathrm{d}t.$$

因为 $f(t)=t^2+t+1$ 是二次多项式，所以当取节点数 $n\geqslant 2$ 时的高斯-切比雪夫求积公式均精确成立.

若用本题(2)的三点高斯-切比雪夫求积公式计算积分 $I[f]$，有

$$I[f]=\int_{-1}^1\frac{t^2+t+1}{\sqrt{1-t^2}}\mathrm{d}t=\frac{\pi}{3}\left[\left(-\frac{\sqrt{3}}{2}\right)^2+\left(-\frac{\sqrt{3}}{2}\right)\right.$$

$$\left.+1+1+\left(\frac{\sqrt{3}}{2}\right)^2+\frac{\sqrt{3}}{2}+1\right]=\frac{3}{2}\pi.$$

例 9 构造形如

$$\int_0^1\sqrt{x}f(x)\mathrm{d}x\approx\omega_0 f(x_0)+\omega_1 f(x_1)$$

的高斯型求积公式.

解 应该指出的是，利用正交多项式的零点构造高斯求积公式，这种方法只是针对某些特殊的权函数才有效，而对于一般的权函数，要构造正交多项式是不容易的，即使有了表达式，求它的根也比较困难. 一般可直接由代数精度的定义来构造.

本题权函数 $\rho(x)=\sqrt{x}$，因为是两个节点高斯型的，所以求积公式的代数精度为 3.

令它对 $f(x)=1,x,x^2,x^3$ 精确成立，得到方程组

$$
\begin{cases}
\omega_0 + \omega_1 = \dfrac{2}{3} \\
x_0\omega_0 + x_1\omega_1 = \dfrac{2}{5} \\
x_0^2\omega_0 + x_1^2\omega_1 = \dfrac{2}{7} \\
x_0^3\omega_0 + x_1^3\omega_1 = \dfrac{2}{9}
\end{cases}
$$

因为它是一个非线性方程组，所以求解比较困难. 由 $x_0\omega_0 + x_1\omega_1 = x_0(\omega_0 + \omega_1) + (x_1 - x_0)\omega_1$，利用方程组的第一个式子，可将第二个式子化为

$$
\frac{2}{3}x_0 + (x_1 - x_0)\omega_1 = \frac{2}{5} \tag{4}
$$

同样地，可利用第二个式子化第三个式子，利用第三式化第四式，分别可得

$$
\frac{2}{5}x_0 + (x_1 - x_0)x_1\omega_1 = \frac{2}{7} \tag{5}
$$

$$
\frac{2}{7}x_0 + (x_1 - x_0)x_1^2\omega_1 = \frac{2}{9} \tag{6}
$$

从式(4)、式(5)和式(6)中消去$(x_1 - x_0)\omega_1$，有

$$
\begin{cases}
\dfrac{2}{5}x_0 + \left(\dfrac{2}{5} - \dfrac{2}{3}x_0\right)x_1 = \dfrac{2}{7} \\
\dfrac{2}{7}x_0 + \left(\dfrac{2}{7} - \dfrac{2}{5}x_0\right)x_1^2 = \dfrac{2}{9}
\end{cases}
$$

进一步整理得

$$
\begin{cases}
\dfrac{2}{5}(x_0 + x_1) - \dfrac{2}{3}x_0x_1 = \dfrac{2}{7} \\
\dfrac{2}{7}(x_0 + x_1) - \dfrac{2}{5}x_0x_1 = \dfrac{2}{9}
\end{cases}
$$

由此解出 $x_0x_1=\frac{5}{21}$，$x_0+x_1=\frac{10}{9}$. 从而可解出

$$x_0=0.821162,\ x_1=0.289949;$$

$$\omega_0=0.389111, \omega_1=0.277556.$$

于是所求高斯型求积公式为

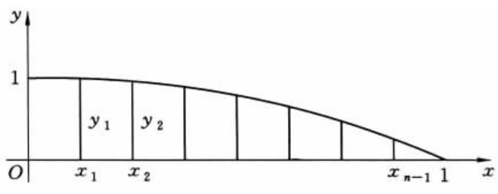
$$\int_0^1 \sqrt{x}f(x)\mathrm{d}x \approx 0.389111f(0.821162)$$

$$+0.277556f(0.289949).$$

例 10　求二阶常微分方程 $y''-(1+x^2)y+1=0$ 满足边界条件 $y(0)=1, y(1)=0$ 的解.

解　利用数值方法求解先进行离散化，只求一些离散点上的函数值.

如图(例)3-1 所示把区间$[0,1]n$ 等分，每个小区间为$[x_i, x_i+1]$，$(i=0,1,\cdots,n-1)$，长度为 $h=\frac{1}{n}$.



图(例)3-1

令 $x_0=0, x_n=1$，则 $x_i=ih$ 对应的函数值为 $y_0, y_1\cdots, y_n$，由题设边界条件 $y_0=1, y_n=0$ 不必求解，只需在中间的 $x_1,\cdots,x_{n-1}$ 这 $n-1$ 个点上建立方程，用数值微分来代替二阶导数，例如，可用二阶差商近似代替方程中的二阶导数，即

$$\frac{y_{i-1}-2y_i+y_{i+1}}{h^2}-(1+x_i^2)y_i=-1$$

整理 $y_{i-1}-[2+(1+x_i^2)h^2]y_i+y_{i+1}=-h^2$,$(i=1,2,\cdots,n-1)$.
令 $\alpha_i=-[2+(1+x_i^2)h^2]$,得方程组

$$\begin{bmatrix} \alpha_1 & 1 & & \\ 1 & \alpha_2 & & \\ & \vdots & & \\ & \alpha_{n-1} & 1 \\ & 1 & \alpha_n \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{bmatrix}=\begin{bmatrix} -h^2 \\ -h^2 \\ \vdots \\ -h^2 \\ -h^2 \end{bmatrix}$$

求解可得 $y_1,y_2,\cdots,y_{n-1}$.

上述解法也常称为常微分方程边值问题的差分解法,也是数值微分公式的应用.

习　题　三

1. 分别用复合梯形公式和复合抛物线公式计算下列积分,并比较结果.

(1) $\int_0^1 \frac{x}{4+x^2}\mathrm{d}x$,　$(n=8)$;

(2) $\int_1^9 \sqrt{x}\mathrm{d}x$,　$(n=4)$.

2. 若用复合梯形公式求 $\int_0^1 \mathrm{e}^{-x}\mathrm{d}x$ 的近似值,问要将积分区间 $[0,1]$ 分成多少份才能保证计算结果有四位有效数字?若用复合辛甫生公式呢?

3. 若 $f''(x)>0$,证明用梯形公式计算积分 $\int_a^b f(x)\mathrm{d}x$ 所得的结果比准确值大,并说明其几何意义.

4. 用代数精确度定义直接验证抛物线求积公式

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right]$$

具有 3 次代数精度．

5．写出 $n=3$ 的牛顿-柯特斯公式，并求出其代数精度，利用此公式计算积分 $\int_0^1 \frac{1}{1+x^2}\mathrm{d}x$ 的近似值．

6．设 $f(x)$ 在 $[a,b]$ 上可积，证明对于积分 $\int_a^b f(x)\mathrm{d}x$ 的复合梯形公式及复合抛物线公式当 $n\to\infty$ 时，收敛于 $\int_a^b f(x)\mathrm{d}x$.

7．确定下列求积公式中的待定系数，使其代数精度尽量高，并指出其所具有的代数精度．

(1) $\int_0^2 f(x)\mathrm{d}x\approx\omega_0 f(0)+\omega_1 f(1)+\omega_2 f(2)$；

(2) $\int_{-2h}^{2h} f(x)\mathrm{d}x\approx\omega_{-1} f(-h)+\omega_0 f(0)+\omega_1 f(h)$；

(3) $\int_0^h f(x)\mathrm{d}x\approx\omega_1 f(0)+\omega_2 f(h)+\omega_3 f'(0)$；

(4) $\int_0^h f(x)\mathrm{d}x\approx\frac{h}{2}[f(0)+f(h)]+ah^2[f'(0)-f'(h)]$.

8．用龙贝格求积算法，计算积分

$$I=\int_0^{\pi} \mathrm{e}^x\cos x\mathrm{d}x.$$

9．试用下述算法计算积分

$$I=\int_1^3 \frac{1}{y}\mathrm{d}y$$

(1) 利用龙贝格求积公式（计算到 R_1 为止）；

(2) 利用三点及五点高斯求积公式计算；

(3) 将积分区间分成四等份，在每一段用两点高斯求积公式，然后累加得 I 的值．（I 之精确值是 1.098 612）.

10. **已知函数** $y=f(x)$**的数值表**

x	1.0	1.1	1.2	1.3	1.4
$f(x)$	0.2500	0.2268	0.2066	0.1890	0.1736

用三点数值微分公式求 $f'(1.0)$，$f'(1.1)$**和** $f'(1.2)$.

第四章　非线性方程的数值解法

在科学研究和工程设计中常常会遇到求解非线性方程

$$f(x)=0 \tag{4.1}$$

的问题，其中 $f(x)$ 为实变量 x 的非线性函数．

一般讲，讨论非线性问题的数值方法，与线性问题相比，无论从理论上还是计算方法上非线性问题都要复杂得多．方程

$$f(x)=0$$

的解通常称为方程的根，或称为函数 $f(x)$ 的零点．

如果 $f(x)$ 是多项式，即

$$f(x)=a_0+a_1x+\cdots+a_nx^n,(a_n\neq 0)$$

则称方程(4.1)为代数方程．

如果 $f(x)$ 中含有三角函数，指数函数或其他超越函数，则称方程(4.1)为超越方程．

例如

$$x^2-5x+2=0 \tag{4.2}$$

$$2^x-5x+2=0 \tag{4.3}$$

方程(4.2)是一个二次代数方程，它的两个根可以表示为有限形式 $x=\frac{5}{2}\pm\frac{1}{2}\sqrt{17}$.

方程(4.3)是一个超越方程，一般而言，必须用数值方法求它的解．

方程的根可能是实数，也可能是复数，相应地称为实根和复根．如果对于 x^* 有

$$f(x^*)=0，但\ f'(x^*)\neq 0$$

则称 x^* 为方程 $f(x)=0$ 的单根；如果有

$$f(x^*) = f'(x^*) = \cdots = f^{(k)}(x^*) = 0$$

但
$$f^{(k+1)}(x^*) \neq 0$$

则称 x^* 为 $f(x)=0$ 的 $k+1$ 重根．

对于高次代数方程，由代数学基本定理知其根（实根或复根）的个数与其次数相同，但对超越方程就复杂得多，如果有解，其解可能是一个或几个，也可能是无穷多个．在大多数情况下，对于高于四次的代数方程及超越方程没有精确的求根公式．事实上，实际应用中也不一定需要得到根的精确表达式，只要得到满足一定精度要求的根的近似值就可以了．

求非线性方程的近似根，一般需要解决下列几个问题：

（1）根的存在性．即方程有没有根？如有根，有几个根？

（2）根的隔离．即找出有根区间，使得在一些较小区间中方程只有一个实根，称为根的隔离，从而能获取根的较粗糙的近似值．一般可通过作图，分析函数或曲线 $y=f(x)$ 的性态，大范围搜索等方法来实现．例如，常用的有"逐步搜索法"，它是从某点 x_0 出发以 h 为步长依次取点 $x_k = x_0 + kh, (k=1,2,\cdots,n)$，如遇有 $f(x_{k-1})f(x_k)<0$，则可断定在区间 $[x_{k-1}, x_k]$ 中连续函数 $f(x)$ 至少有一个零点，也即确定了方程 $f(x)=0$ 的一个有根区间．此时可取 x_{k-1} 或 x_k 作为根的近似值．只含有一个实根的区间称为隔根区间．

（3）根的精确化．即已知一个根的近似值后，设法逐步把根精确化，直到满足精度为止．

本章将介绍几种计算机上常用的有效的求解一般非线性方程（4.1）的数值方法．

4.1 二分法

设 $f(x)$ 在区间 $[a,b]$ 上连续，且 $f(a)f(b)<0$，那么，根据连续函数的零点定理，方程 $f(x)=0$ 在 (a,b) 内至少有一实根．为

简单起见，不妨设 $f(x)=0$ 在(a,b)内只有一个实根 x^*.

用二分法求实根 x^* 的近似值的基本思想，就是逐步将含有根 x^* 的区间二分，通过判断函数值的符号，逐步对半缩小有根区间，直到区间缩小到容许误差范围之内，然后取区间的中点为根 x^* 的近似值．其具体做法如下：

为了便于讨论，不妨设 $f(a)<0$，$f(b)>0$，如图 4-1 所示．

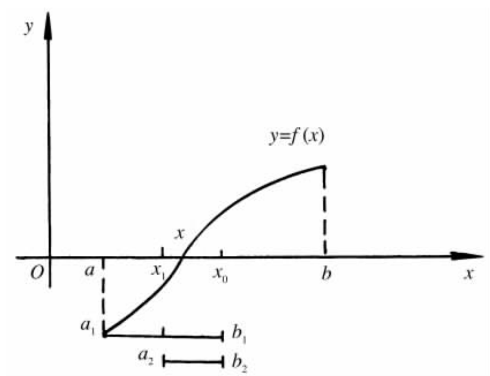


图 4-1

取区间$[a,b]$的中点 $x_0=\frac{1}{2}(a+b)$，计算函数值 $f(x_0)$，如果恰好 $f(x_0)=0$，则我们已经找到方程的根 $x^*=\frac{1}{2}(a+b)$．否则 $f(x_0)$或与 $f(a)$异号，或与 $f(b)$异号．若 $f(x_0)>0$，则记 $a_1=a$，$b_1=\frac{1}{2}(a+b)$（即只取原来区间的左半部分）；若 $f(x_0)<0$，则记 $a_1=\frac{1}{2}(a+b)$，$b_1=b$（即只取原来区间的右半部分）．则区间$[a_1,b_1]$是方程新的有根区间，它被包含在旧的有根区间$[a,b]$内，且其长度仅是$[a,b]$的一半，对缩小了的有根区间$[a_1,b_1]$，又可施行同样手续，即用中点 $x_1=\frac{a_1+b_1}{2}$将$[a_1,b_1]$再分为两半，然后判

断 x_1 是不是根，若不是，则用 $f(a_1)f(x_1)<0$ 是否成立判断所求根 x^* 在 x_1 的哪一侧，从而又确定一个新的有根区间$[a_2,b_2]$，其长度是$[a_1,b_1]$的一半，如此反复二分下去，重复上述同样过程便可得到一系列不断缩小的有根区间：

$$[a,b],\ [a_1,b_1],\ [a_2,b_2],\ \cdots,\ [a_k,\ b_k],\cdots$$

其中，每个区间落在前一个区间内，且长度都是前一区间的一半，从而区间$[a_k,b_k]$的长度为

$$b_k-a_k=\frac{1}{2}(b_{k-1}-a_{k-1})=\cdots=\frac{1}{2^k}(b-a)$$

每次二分后，若取有根区间$[a_k,\ b_k]$的中点 $x_k=\frac{1}{2}(a_k+b_k)$作为 $f(x)=0$ 的根的近似值，则在二分过程中可以获得一个近似根序列

$$x_0,\ x_1,\ x_2,\ \cdots,\ x_k,\ \cdots$$

显然，如果二分过程无限地继续下去，该序列必以方程的根 x^* 为极限，即有$\lim\limits_{k\to\infty}x_k=x^*$.

但在实际计算时，不可能也没有必要完成这个无限过程，只要有根区间的长度满足要求的精度，计算就可停止，由于

$$|x^*-x_k|\leqslant\frac{1}{2}(b_k-a_k)=\frac{1}{2^{k+1}}(b-a)$$

故对于预先给定的误差 $\varepsilon>0$，只要二分次数满足不等式

$$\frac{1}{2^{k+1}}(b-a)<\varepsilon$$

则 x_k 就是方程 $f(x)=0$ 的一个满足给定精度的近似根了.

上述求非线性方程实根近似值的方法称为二分法.

例 4.1 求方程 $f(x)=x^3-2x-5=0$ 在区间$[2,3]$内的实根近似值，并指出其误差.

解 用二分法．这里 $a=2$，$b=3$，且 $f(a)=-1<0$，$f(b)=16>0$，$f(x)=x^3-2x-5$ 在 $[2,3]$ 上连续，所以 $[2,3]$ 是 $f(x)$ 的有根区间．

用上述二分法计算结果见表 4-1，若取根的近似值为 $x^* \approx x_6=2.1015625$，则其误差为

$$|x^*-x_6| \leqslant \frac{1}{2^7}(3-2)=0.078125$$

（可求得根的精确值 $x^*=2.0945515$）．

表 4-1 计算结果

k	a_k	b_k	x_k	$f(x_h)$的符号
0	2	3	2.5	+
1	2	2.5	2.25	+
2	2	2.25	2.125	+
3	2	2.125	2.062 5	−
4	2.062 5	2.125	2.093 75	−
5	2.093 75	2.125	2.109 375	+
6	2.093 75	2.109 375	2.101 562 5	

二分法的优点是方法简单、编制程序容易，且对函数 $f(x)$ 性质要求不高，仅仅要求它连续且在区间两端点的函数值异号，收敛速度与比值为 1/2 的等比级数相同．但二分法只能用于求实函数的实根，不能用于求复根及偶数重根．

4.2 迭代法

迭代法是数值计算中的一类重要方法,它是求解方程或方程组的一类基本方法,在科学与工程计算中经常使用,尤其是由于计算机的普遍使用,使迭代法的应用更为广泛.

一、迭代法的基本思想

迭代法是一种逐次逼近的方法,它的基本思想是使用某个固定公式反复校正根的近似值,从而得到一个近似根序列$\{x_k\}$,使得该序列的极限就是方程的根.

对于非线性方程 $f(x)=0$,用迭代法求根的具体做法是:先将方程 $f(x)=0$ 改写成便于迭代的等价形式

$$x=\varphi(x) \tag{4.4}$$

例如,最简单的可取 $\varphi(x)=x+f(x)$,然后取初始值 x_0,计算

$$x_{k+1}=\varphi(x_k), \quad (k=0,1,2,\cdots) \tag{4.5}$$

可得一序列$\{x_k\}$. 显然,如果 $\varphi(x)$连续,且序列$\{x_k\}$收敛到 x^*,则有

$$x^*=\lim_{k\to\infty}x_{k+1}=\lim_{k\to\infty}\varphi(x_k)=\varphi(\lim_{k\to\infty}x_k)=\varphi(x^*)$$

即 x^* 满足方程 $x=\varphi(x)$,由于方程 $x=\varphi(x)$与 $f(x)=0$ 等价,所以 x^* 即为方程 $f(x)=0$ 的一个根. 此时称该迭代法收敛;否则称为发散.

上述这种迭代法,是从一个初始近似值 x_0 出发计算迭代的,一般也称为单点迭代法,$\varphi(x)$称为迭代函数,由式(4.5)产生的数列$\{x_k\}$称为迭代序列. 显然,$\varphi(x)$依赖于函数 $f(x)$,用不同的方法构造迭代函数就得到不同的迭代方法.

但是要注意,迭代法的效果并不总是令人满意的,迭代过程可

能收敛,也可能发散,就是同一个方程,当取不同迭代格式时也会得到完全不同的效果.

例 4.2 用迭代法求方程 $f(x)=x^2+x-16=0$ 的根(已知方程的一个根是 $x^*=3.5311289$).

解 把方程分别改写成下列三种等价的便于迭代的形式:

(1) $x=16-x^2$;

(2) $x=\dfrac{16}{x+1}$;

(3) $x=x-\dfrac{x^2+x-16}{2x+1}$.

若取相同的初值 $x_0=3$,经分别迭代计算得到的结果列于表 4-2 中.

表 4-2　　　　计算结果

k	0	1	2	3	4	5	6	7	8	9
$x_{(1)}$	3	7	−33	−1 073	−115 131 3					
$x_{(2)}$	3	4	3.2	3.810	3.326	3.699	3.405	3.632	3.454	3.592
$x_{(3)}$	3	3.571	3.531							

从表 4-2 中数据的变化趋势看,迭代格式(1)得到的序列 $\{x_{(1)}\}$ 是发散的;迭代格式(2)得到的序列 $\{x_{(2)}\}$ 虽然收敛,但收敛速度很慢;迭代格式(3)得到的序列 $\{x_{(3)}\}$ 不但收敛,而且收敛速度很快.

从上述计算可以看出,迭代函数 $\varphi(x)$ 选取不同,相应的迭代序列 $\{x_k\}$ 的收敛情况也不一样,即使同是收敛的迭代格式,也有一个速度快慢的问题. 对于一个发散的迭代过程,纵使进行千百次迭代,其结果也是毫无价值的. 迭代法的敛散性关键取决于迭代函数 $\varphi(x)$ 在有根区间内的性态.

首先,$\varphi(x)$必须使初始值 x_0 产生的$\{x_k\}\subseteq[a,b]$($\varphi(x)$的定义域). 其次,从迭代法的几何意义图 4-2 看到,曲线 $\varphi(x)$变化比较缓慢是迭代收敛的良好征兆. 图中(1),(2)是收敛的,其共同特征是曲线 $y=\varphi(x)$走势平坦;而(3),(4)是发散的,其共同特征是曲线 $y=\varphi(x)$走势很陡.

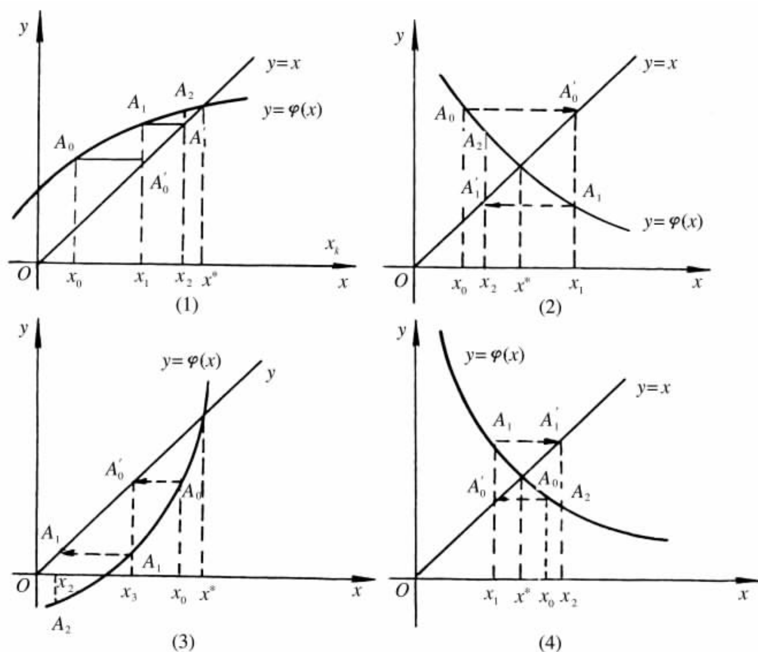


图 4-2

二、迭代法的收敛性

设方程 $f(x)=0$ 的迭代格式为

$$x_{k+1}=\varphi(x_k),\ (k=0,1,2,\cdots),x\in[a,b]$$

下面给出一个迭代法收敛的充分条件以及两个误差估计式.

定理 4.1 (收敛性定理)

设迭代函数 $\varphi(x)$ 在 $[a,b]$ 上具有连续的一阶导数，且

（1）当 $x\in[a,b]$ 时，$\varphi(x)\in[a,b]$；

（2）存在正数 $L<1$，使对任意 $x\in[a,b]$，有

$$|\varphi'(x)|\leqslant L<1$$

成立，则

（1）方程 $x=\varphi(x)$ 在 $[a,b]$ 上有唯一的根 x^*，且对任取初始近似值 $x_0\in[a,b]$，迭代法 $x_{k+1}=\varphi(x_k)$，$(k=0,1,\cdots)$ 均收敛，并有 $\lim\limits_{k\to\infty}x_k=x^*$；

$$(2)\ |x^*-x_k|\leqslant\frac{1}{1-L}|x_{k+1}-x_k|; \tag{4.6}$$

$$(3)\ |x^*-x_k|\leqslant\frac{L^k}{1-L}|x_1-x_0|. \tag{4.7}$$

证明　（1）先证 x^* 的存在性．由于在 $[a,b]$ 上 $\varphi'(x)$ 存在，所以 $\varphi(x)$ 连续，作 $g(x)=x-\varphi(x)$，则 $g(x)$ 在 $[a,b]$ 上也连续．由条件（1）有

$$g(a)=a-\varphi(a)\leqslant 0,\quad g(b)=b-\varphi(b)\geqslant 0$$

故必存在 $x^*\in[a,b]$，使 $g(x^*)=0$，即 $x^*=\varphi(x^*)$.

再证 x^* 的唯一性．若方程 $x=\varphi(x)$ 在 $[a,b]$ 上有两个实根 x^* 和 $\bar{x}^*$，则由微分中值定理及条件（2）知

$$\begin{aligned}|x^*-\bar{x}^*|&=|\varphi(x^*)-\varphi(\bar{x}^*)|\\&=|\varphi'(\xi)||x^*-\bar{x}^*|\leqslant L|x^*-\bar{x}^*|\end{aligned}$$

（其中 ξ 在 x^* 与 $\bar{x}^*$ 之间，$L<1$），显然，上式只有在 $x^*=\bar{x}^*$ 时成立．

最后证明迭代法的收敛性．条件（1）表明，当 $x_0\in[a,b]$ 时，$x_k\in[a,b]$，$(k=0,1,\cdots)$．因为

$$x^*-x_{k+1}=\varphi(x^*)-\varphi(x_k)=\varphi'(\xi_k)(x^*-x_k)$$

（其中，ξ_k 介于 x^* 与 x_k 之间）

故由条件(1)知

$$|x^*-x_{k+1}|\leqslant L|x^*-x_k|,(k=0,1,2,\cdots) \qquad (4.8)$$

应用不等式(4.8),由递推可得

$$\begin{aligned}0\leqslant|x^*-x_k|&\leqslant L|x^*-x_{k-1}|\\&\leqslant L^2|x^*-x_{k-2}|\leqslant\cdots\leqslant L^k|x^*-x_0|,\end{aligned}$$

因为 $L<1$,所以有$\lim\limits_{k\to\infty}L^k|x^*-x_0|=0$,从而有

$$\lim_{k\to\infty}x_k=x^*.$$

(2) 由不等式(4.8)可得

$$\begin{aligned}|x_{k+1}-x_k|&=|(x^*-x_k)-(x^*-x_{k+1})|\\&\geqslant|x^*-x_k|-|x^*-x_{k+1}|\\&\geqslant|x^*-x_k|-L|x^*-x_k|\\&=(1-L)|x^*-x_k|\end{aligned}$$

从而 $$|x^*-x_k|\leqslant\frac{1}{1-L}|x_{k+1}-x_k|,(k=0,1,2\cdots)$$

即误差估计式(4.6)成立.

(3) 由于

$$\begin{aligned}|x_{k+1}-x_k|&=|\varphi(x_k)-\varphi(x_{k-1})|\\&=|\varphi'(\xi_k)(x_k-x_{k-1})|\\&\leqslant L|x_k-x_{k-1}|\end{aligned}$$

(其中,ξ_k 在 x_k 和 x_{k-1} 之间,$k=1,2,\cdots$) (4.9)

于是由误差估计式(4.6)且应用式(4.9),有

$$\begin{aligned}|x^*-x_k|&\leqslant\frac{1}{1-L}|x_{k+1}-x_k|\leqslant\frac{L}{1-L}|x_k-x_{k-1}|\\&\leqslant\frac{L^2}{1-L}|x_{k-1}-x_{k-2}|\leqslant\cdots\leqslant\frac{L^k}{1-L}|x_1-x_0|\end{aligned}$$

即误差估计式(4.7)成立.

误差估计式(4.6)说明,只要相邻两次迭代值 x_k, x_{k+1} 的差 $|x_k-x_{k+1}|$充分小,就可以保证迭代值 x_k 足够精确,所以常用条件

$$|x_k - x_{k-1}| < \varepsilon$$

来控制迭代过程是否结束,当上述条件满足时就停止迭代,且取 $x^* \approx x_k$ 为所求根的满足精度要求的近似值. 不过当 $L \approx 1$ 时,这个方法就不可靠了.

误差估计式(4.7)还可用来确定使误差达到给定精度所需迭代的次数,即,若事先给出计算精度 ε,要求 $|x^* - x_k| < \varepsilon$,则由式(4.7):

$$|x^* - x_k| \leqslant \frac{L^k}{1-L}|x_1 - x_0| < \varepsilon$$

就可确定迭代次数应取

$$k > \left[\ln \frac{\varepsilon(1-L)}{|x_1 - x_0|} / \ln L\right]$$

但要注意,定理 4.1 中给出的保证收敛的条件 $|\varphi'(x)| \leqslant L < 1$,对于较大范围的含根区间,有时可能不成立. 所以,实际使用迭代法时,常在根 x^* 邻近范围内考虑. 因为,如果 $\varphi'(x)$ 在根 x^* 的某邻域连续,且 $|\varphi'(x^*)| < 1$,那么由连续函数的性质知,必存在 x^* 的一个邻域 $s = \{x \mid |x - x^*| < \delta\}$,当 $x \in s$ 时,不等式 $|\varphi'(x)| \leqslant L < 1$ 成立. 且有

$$|\varphi(x) - x^*| = |\varphi(x) - \varphi(x^*)| \leqslant L|x - x^*| < |x - x^*| < \delta$$

即对任意 $x \in s$,有 $\varphi(x) \in s$. 于是由定理 4.1 可知对任意选取的初值 $x_0 \in s$,迭代过程 $x_{k+1} = \varphi(x_k)$ 都收敛. 也即只要构造迭代函数,使其在根的邻近满足导数的绝对值有小于 1 的上界,即可保证迭代法收敛.

这种在根的邻域具有的收敛性称作局部收敛性.

例 4.3 方程 $x = e^{-x}$ 有唯一实根 $x^* \in (0,1)$,试讨论迭代法 $x_{k+1} = e^{-x_k}$,$(k = 0,1,2,\cdots)$ 的收敛性.

解 这里迭代函数为 $\varphi(x) = e^{-x}$, $\varphi'(x) = -e^{-x}$,显然,在区间(0,1)内 $\varphi'(x)$ 连续且有 $|\varphi'(x)| = e^{-x} < e^0 = 1$

由上述局部收敛性讨论知迭代法 $x_{k+1} = e^{-x_k}$ 在根 x^* 附近具

有局部收敛性，因此只要 x_0 取得充分接近 x^*，则迭代必收敛．如取 $x_0=0.5$，用此迭代格式迭代九次可得到精确到小数后三位的近似值 $x_9=0.567$.

迭代法最显著的优点是算法的逻辑结构简单，所以相应计算程序比较简单，特别适合在计算机上计算．

4.3　迭代法的收敛阶和加速收敛方法

一、迭代法的收敛阶

我们已知迭代法依靠收敛的迭代序列来求方程根的近似值．一个迭代法要具有实用的价值，首先要求它是收敛的，其次还要求它收敛得比较快．不同的迭代格式所得到的迭代序列即使都收敛，也会有快慢之分，即存在一个收敛速度的问题．所谓迭代过程的收敛速度，是指在接近收敛时迭代误差的下降速度，一般地，它主要由方法决定，方程的性态也会起一些影响．

用什么来反映迭代序列的收敛速度呢？下面我们引进迭代法的收敛阶的概念，这是迭代法的一个重要概念，它反映了迭代序列的收敛速度，是衡量一个迭代法好坏的标志之一．

定义 4.1　设序列 $\{x_k\}$ 是收敛于方程 $f(x)=0$ 的根 x^* 的迭代序列，即 $x^*=\lim\limits_{k\to\infty}x_k$. 记 $e_k=x_k-x^*$，$(k=0,1,2,\cdots)$，表示各步的迭代误差．

若存在常数 $P(P\geqslant 1)$ 和 $c(c>0)$，使得

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^P}=c$$

则称序列 $\{x_k\}$ 是 P 阶收敛的．特别地，当 $P=1$ 且 $0<c<1$ 时称为线性收敛；$P>1$ 时称为超线性收敛；$P=2$ 时称为平方收敛．

如果由迭代函数 $\varphi(x)$ 产生的序列 $\{x_k\}$ 是 P 阶收敛，则称 $\varphi(x)$ 是 P 阶迭代函数，并称迭代法 $x_{k+1}=\varphi(x_k)$ 是 P 阶收敛．

显然，数 P 的大小反映了迭代法收敛速度的快慢，P 越大，则收敛越快，所以迭代法的收敛阶是对迭代法收敛速度的一种度量.

由上述定义可以推得以下结论：

定理 4.2 对于迭代过程 $x_{k+1}=\varphi(x_k)$，如果迭代函数 $\varphi(x)$ 在所求根 x^* 的邻近有连续的二阶导数，且

$$|\varphi'(x^*)|<1$$

则有

(1) 当 $\varphi'(x^*)\neq 0$ 时，迭代过程为线性收敛；

(2) 当 $\varphi'(x^*)=0$，而 $\varphi''(x^*)\neq 0$ 时迭代过程为平方收敛.

证明 由已知条件，迭代过程具有局部收敛性，现分别证明上述结论.

(1) 对于在根 x^* 邻近收敛的迭代公式 $x_{k+1}=\varphi(x_k)$，由于

$$\begin{aligned}e_{k+1}&=x_{k+1}-x^*=\varphi(x_k)-\varphi(x^*)\\&=\varphi'(\xi)(x_k-x^*)=\varphi'(\xi)e_k\end{aligned}$$

这里 ξ 为 x_k 与 x^* 之间的某一点，因而当 x_k 在根 x^* 的附近时，将有

$$e_{k+1}\approx\varphi'(x^*)e_k$$

即

$$\frac{e_{k+1}}{e_k}\longrightarrow\varphi'(x^*),\quad(k\to\infty)$$

这样，若 $\varphi'(x^*)\neq 0$，则该迭代过程仅为线性收敛.

(2) 若 $\varphi'(x^*)=0$，将 $\varphi(x_k)$ 在根 x^* 处作泰勒展开，注意条件 $\varphi'(x^*)=0$，有

$$\varphi(x_k)=\varphi(x^*)+\frac{\varphi''(\xi)}{2!}(x_k-x^*)^2$$

而 $\varphi(x_k)=x_{k+1}$，$\varphi(x^*)=x^*$，由上式得

$$x_{k+1}-x^*=\frac{\varphi''(\xi)}{2!}(x_k-x^*)^2$$

因此,对迭代误差,当 $k\to\infty$时,有

$$\frac{e_{k+1}}{e_k^2}\longrightarrow\frac{\varphi''(x^*)}{2}$$

这表明迭代过程为平方收敛.

上述定理告诉我们,迭代过程的收敛速度取决于迭代函数 $\varphi(x)$的选取.如果当 $x\in[a,b]$时,$\varphi'(x)\neq 0$,则该迭代过程只可能是线性收敛.

显然,要达到同样的计算精度,高阶算法要比低阶算法特别是线性算法步数少得多,所以收敛阶是衡量算法的重要标准.

二、加速收敛法

对于一个收敛的迭代过程,只要迭代次数足够多,从理论上讲就能得到满足任意精度的结果.但这里有一个重要的问题——收敛速度问题,若迭代过程的收敛速度太慢,则计算工作量就很大,因而有必要研究加速迭代收敛性的方法.

从定理 4.2 知,当迭代函数 $\varphi(x)$在根 x^* 处有 $\varphi'(x^*)\neq 0$ 时,迭代过程仅为线性收敛,下面介绍一种对于一般的线性收敛情况的加速方法.

设 x_k 是方程根 x^* 的某个近似值,则 $x_k-x^*\neq 0$,且相继两个迭代值为

$$\begin{cases}x_{k+1}=\varphi(x_k)\\ x_{k+2}=\varphi(x_{k+1})\end{cases}$$

由中值定理可得

$$\begin{cases}x^*-x_{k+1}=\varphi'(\xi_k)(x^*-x_k)\\ x^*-x_{k+2}=\varphi'(\xi_{k+1})(x^*-x_{k+1})\end{cases}$$

其中 ξ_k 在 x_k 与 x^* 之间，ξ_{k+1} 在 x_{k+1} 与 x^* 之间，假定 $\varphi'(x)$ 在 x 变化时改变不大，可令 $\varphi'(x)\approx L$，于是可得，当 k 充分大时，有

$$\frac{x^*-x_{k+1}}{x^*-x_k}\approx\frac{x^*-x_{k+2}}{x^*-x_{k+1}} \tag{4.10}$$

由此解出
$$x^*\approx x_{k+2}-\frac{(x_{k+2}-x_{k+1})^2}{x_{k+2}-2x_{k+1}+x_k} \tag{4.11}$$

这样又获得了一个由 x_k, x_{k+1}, x_{k+2} 确定的新的近似值，只要 k 充分大，使式(4.10)得到较好满足，那么这个新的近似值就有可能比 x_{k+2} 更好地逼近 x^*，由此在收敛速度较慢的序列 $\{x_k\}$ 基础上，通过算式

$$\bar{\bar{x}}_k=x_{k+2}-\frac{(x_{k+2}-x_{k+1})^2}{x_{k+2}-2x_{k+1}+x_k}$$

有可能产生一个收敛速度较快的新序列 $\{\bar{\bar{x}}\}$，这种加速收敛的方法称为埃特金(Aitken)加速方法.

若把式(4.11)中的 x_{k+1} 记作 $\tilde{x}_{k+1}$，把 x_{k+2} 记作 $\bar{x}_{k+1}$，并把式(4.11)右端计算的值作为新的 x_{k+1}，那么，从 x_k 到这个新的 x_{k+1} 就算作迭代了一次，则埃特金方法的计算公式如下：

$$\begin{cases}\tilde{x}_{k+1}=\varphi(x_k) & \text{(迭代)}\\ \bar{x}_{k+1}=\varphi(\tilde{x}_{k+1}) & \text{(迭代)}\\ x_{k+1}=\bar{x}_{k+1}-\dfrac{(\bar{x}_{k+1}-\tilde{x}_{k+1})^2}{\bar{x}_{k+1}-2\tilde{x}_{k+1}+x_k} & \text{(加速)}\end{cases}$$

$$(k=0,1,2,\cdots) \tag{4.12}$$

也即每一步迭代要计算两个 $\varphi(x)$ 的值，再进行校正. 可以证明它是一个二阶收敛的方法，一般可以用来加速具有线性收敛序列的收敛速度.

例 4.4 用埃特金加速方法求方程 $x=\mathrm{e}^{-x}$ 在 $x_0=0.5$ 附近的根 x^* 的近似值.

解　用埃特金方法的计算公式为

$$\begin{cases} \tilde{x}_{k+1}=\mathrm{e}^{-x_k} \\ \bar{x}_{k+1}=\mathrm{e}^{-\tilde{x}_{k+1}} \\ x_{k+1}=\bar{x}_{k+1}-\dfrac{(\bar{x}_{k+1}-\tilde{x}_{k+1})^2}{\bar{x}_{k+1}-2\tilde{x}_{k+1}+x_k} \end{cases} \quad (k=0,1,2,\cdots)$$

计算结果为：$x_0=0.5$，$\tilde{x}_1=0.606\,53$，$\bar{x}_1=0.545\,24$，$x_1=0.567\,62$；

$\tilde{x}_2=0.566\,87$，$\bar{x}_2=0.567\,30$，$x_2=0.567\,14$，

这里只迭代两步得到的 x^* 的近似值 x_2 已同于前面例 4.3 迭代 18 次才得到的近似值，可见埃特金方法加速收敛的效果是明显的.

4.4　牛顿迭代法

牛顿迭代法是求解非线性方程 $f(x)=0$ 的一种重要和常用的迭代法，它的基本思想是将非线性函数 $f(x)$ 逐步线性化，从而将非线性方程 $f(x)=0$ 近似地转化为线性方程求解.

一、牛顿法的迭代公式

设已知方程 $f(x)=0$ 的根 x^* 的一个近似值 x_k，把函数 $f(x)$ 在 x_k 处作泰勒展开：

$$f(x)=f(x_k)+f'(x_k)(x-x_k)+\frac{f''(\xi)}{2!}(x-x_k)^2$$

取其前两项近似代替 $f(x)$（称为 $f(x)$ 的线性化），即用线性方程

$f(x_k)+f'(x_k)(x-x_k)=0$　近似方程　$f(x)=0$.

若 $f'(x_k)\neq 0$，则用线性方程的根作为非线性方程 $f(x)=0$ 的新近似根 x_{k+1}，则得牛顿迭代法的计算公式：

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}, \quad (k=0,1,2,\cdots) \tag{4.13}$$

它对应的方程

$$x=x-\frac{f(x)}{f'(x)}, \quad (f'(x)\neq 0)$$

显然和方程 $f(x)=0$ 的等价,所以迭代函数就是

$$\varphi(x)=x-\frac{f(x)}{f'(x)}$$

由 4.2 节的关于收敛的讨论知,由于现在

$$\varphi'(x)=\frac{f(x)f''(x)}{[f'(x)]^2}$$

如果 x^* 是方程 $f(x)=0$ 的一个单根,即 $f(x^*)=0$,而 $f'(x^*)\neq 0$,则 $\varphi'(x^*)=0$,可见在单根 x^* 的附近,对于任意的初值 x_0,由公式(4.13)得到的迭代序列都收敛于根 x^*,而且收敛的速度很快.迭代式(4.13)所确定的求解 $f(x)=0$ 的方法就称为牛顿法.4.2 节的例 4.2 的迭代格式(3)就是一个牛顿迭代格式.

牛顿法有明显的几何意义.方程 $f(x)=0$ 的根 x^* 在几何上表示曲线 $y=f(x)$ 与 x 轴的交点.当我们求得 x^* 的近似值 x_k 以后,过曲线 $y=f(x)$ 上对应点 $(x_k,f(x_k))$ 作 $f(x)$ 的切线,其切线方程为

$$y-f(x_k)=f'(x_k)(x-x_k)$$

则该切线与 x 轴交点横坐标正好是 $x_k-\frac{f(x_k)}{f'(x_k)}$,这就是牛顿迭代公式(4.13)的计算结果.继续取点 $(x_{k+1}, f(x_{k+1}))$,再作曲线 $y=f(x)$ 的切线与 x 相交,又可得 x_{k+2},…,由图 4-3 可知,只要初值取得充分靠近根 x^*,点列 $\{x_k\}$ 就会很快收敛于 x^*.正因为牛顿法有这一明显的几何意义,所以牛顿法也称为切线法.

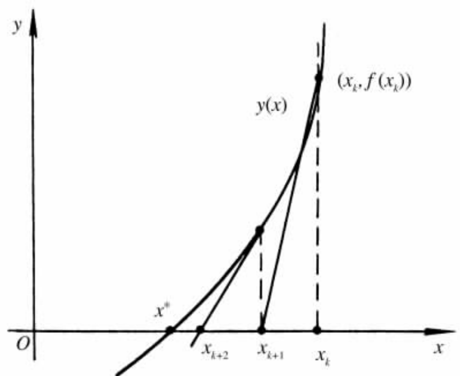


图 4-3

二、牛顿法的收敛性

从上面的讨论已知，对于方程 $f(x)=0$，如果 $f(x)$ 在根 x^* 邻近具有连续的二阶导数，且 x^* 是 $f(x)=0$ 的一个单根，则在根 x^* 的附近，对于任意的初始近似值 x_0，由牛顿迭代法产生的序列 $\{x_k\}$ 收敛于 x^*. 所以牛顿法具有局部收敛性，并且可以进一步证明牛顿迭代法在单根 x^* 附近是平方收敛的，即具有二阶收敛速度. 但是，如果 x^* 是 $f(x)=0$ 的重根时，则可以证明牛顿法仅有线性收敛速度，且重数越高收敛越慢.

牛顿迭代法对初值 x_0 的选取要求比较高，x_0 必须充分靠近 x^* 才能保证局部收敛. 为保证牛顿法的非局部收敛，下面给出一个判断牛顿法的非局部收敛的充分定理：

定理 4.3 设函数 $f(x)$ 在区间 $[a,b]$ 上存在二阶导数，且满足条件：

(1) $f(a)f(b)<0$；

(2) $f'(x)$ 在 $[a,b]$ 上不等于零；

(3) $f''(x)$ 在 $[a,b]$ 上不变号；

(4) 在$[a,b]$上任意选取满足条件 $f(x_0)f''(x_0)>0$ 的初始近似值 x_0.

则由牛顿法产生的序列$\{x_k\}$单调收敛于方程 $f(x)=0$ 在$[a,b]$上的唯一根.

此定理证明较繁,这里从略. 但可给出一些几何解释. 定理中条件(1)保证了方程 $f(x)=0$ 在$[a,b]$上至少存在一个根 x^*;条件(2)表明函数 $f(x)$是单调函数,因此方程的根 x^* 是唯一的;条件(3)保证了 $f(x)$的图形在$[a,b]$上的凹向不改变;条件(4)保证了当 $x_k\in[a,b]$时,有 $x_{k+1}=\varphi(x_k)\in[a,b]$. 如图 4-3 中 $f''(x)>0,f'(x)>0$,图 4-3 所示的 $f(x)$是满足定理条件的,从而由牛顿迭代法的几何意义可明确断定由牛顿法产生的迭代序列$\{x_k\}$是收敛的.

例 4.5 用牛顿迭代法求方程 $x=\mathrm{e}^{-x}$在 $x_0=0.5$ 附近的一个实根.

解 令 $f(x)=x-\mathrm{e}^{-x}$,则此方程的牛顿迭代公式为

$$x_{k+1}=x_k-\frac{x_k-\mathrm{e}^{-x_k}}{1+x_k},\quad (k=0,1,\cdots)$$

取 $x_0=0.5$,计算$\{x_k\}$的有五位小数的前几个值,列于表 4-3.

表 4-3 **计算结果**

k	0	1	2	3
x_k	0.500 00	0.571 02	0.567 16	0.567 14

与例 4.3,例 4.4 的计算结果比较,牛顿法的收敛速度是快的.

例 4.6 对于给定的正数 a,用牛顿法建立求平方根$\sqrt{a}$的收敛的迭代公式.

解 令 $f(x)=x^2-a,(x>0)$,则 $f(x)=0$ 的正根就是$\sqrt{a}$. 用牛顿法求解的迭代公式是

$$x_{k+1}=x_k-\frac{x_k^2-a}{2x_k}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right),\quad (k=0,1,2,\cdots)\tag{4.14}$$

由于当 $x>0$ 时，$f'(x)=2x>0$，$f''(x)=2>0$，故由收敛定理 4.3 知，对于任取满足条件 $x_0>\sqrt{a}$ 的初始近似值，由迭代公式(4.14)所产生的序列 $\{x_k\}$ 必收敛于平方根 $\sqrt{a}$. 公式(4.14)是计算平方根的准确而有效的计算方法.

三、牛顿下山法

由牛顿法的收敛性定理知，牛顿法对初始值 x_0 的选取要求是很高的. 一般地说，牛顿法只有局部收敛性. 当初始值取得离根太远时，迭代将不收敛，而一旦 x_k 进入收敛域内，牛顿法就有平方收敛的速度. 为了扬长避短，扩大初始值选取的范围，下面介绍牛顿法的一种改进——牛顿下山法.

将牛顿法的迭代公式(4.13)修改为

$$x_{k+1}=x_k-\lambda\frac{f(x_k)}{f'(x_k)},\ (k=0,1,2,\cdots) \tag{4.15}$$

其中，λ 是一个参数，λ 的选取应使

$$|f(x_{k+1})|<|f(x_k)|$$

成立，当 $|f(x_{k+1})|<\varepsilon_1$ 或 $|x_{k+1}-x_k|<\varepsilon_2$ 时(其中 $\varepsilon_1,\varepsilon_2$ 为事先给定的精度，称 ε_1 为残量精确度，ε_2 为根的误差限)，就停止迭代，且取 $x^*\approx x_{k+1}$；否则再减小 λ，继续迭代.

按上述迭代过程计算，实际上得到一个以零为下界的严格单调下降的函数值序列 $\{|f(x_k)|\}$. 这个方法就称为牛顿下山法. λ 称为下山因子，要求满足 $0<\varepsilon_\lambda\leqslant\lambda\leqslant1$，$\varepsilon_\lambda$ 为下山因子下界，为了方便，一般开始时可简单地取 $\lambda=1$，然后逐步分半减小，即可选取 $\lambda=1,\ \frac{1}{2},\ \frac{1}{2^2},\cdots,\lambda\geqslant\varepsilon_\lambda$，且使 $|f(x_{k+1})|<|f(x_k)|$.

牛顿下山法计算步骤可归纳如下：

(1) 选取初始近似值 x_0；

(2) 取下山因子 $\lambda=1$；

(3) 计算 x_{k+1}： $x_{k+1}=x_k-\lambda\dfrac{f(x_k)}{f'(x_k)}$；

(4) 计算 $f(x_{k+1})$，并比较 $|f(x_{k+1})|$ 与 $|f(x_k)|$ 的大小，分以下两种情况：

① 若 $|f(x_{k+1})|<|f(x_k)|$，则当 $|x_{k+1}-x_k|<\varepsilon_2$ 时，则就取 $x^*\approx x_{k+1}$，计算过程结束；

当 $|x_{k+1}-x_k|\geqslant\varepsilon_2$ 时，则把 x_{k+1} 作为新的 x_k 值，并重复回到(3)．

② 若 $|f(x_{k+1})|\geqslant|f(x_k)|$，则当 $\lambda\leqslant\varepsilon_\lambda$ 且 $|f(x_{k+1})|<\varepsilon_1$，就取 $x^*\approx x_k$，计算过程结束；否则，若 $\lambda\leqslant\varepsilon_\lambda$，而 $|f(x_{k+1})|\geqslant\varepsilon_1$ 时，则把 x_{k+1} 加上一个适当选定的小正数，即取 $x_{k+1}+\delta$ 作为新的 x_k 值，并转向(3)重复计算；当 $\lambda>\varepsilon_\lambda$，且 $|f(x_{k+1})|\geqslant\varepsilon_1$，则将下山因子缩小一半，取 $\lambda/2$ 代入，并转向(3)重复计算．

牛顿下山法不但放宽了初值 x_0 的选取，且有时对某一初值，虽然用牛顿法不收敛，但用牛顿下山法却可能收敛．

例如，已知方程 $f(x)=x^3-x-1=0$ 的一个根为 $x^*=1.324\,72$，若取初值 $x_0=0.6$，用牛顿法 $x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$ 计算得的第一次近似值 $x_1=17.9$ 反而比 $x_0=0.6$ 更偏离根 x^*．若改用牛顿下山法

$$x_{k+1}=x_k-\lambda\frac{f(x_k)}{f'(x_k)},(k=0,1,2,\cdots) \tag{4.15}$$

计算，仍取 $x_0=0.6$，并令 $\bar{x}_1=17.9$，则从 $\lambda=1$ 开始逐次搜索，当取下山因子 $\lambda=\dfrac{1}{2^5}=\dfrac{1}{32}$时，由公式(4.15)得到修正了的 $x_1=\dfrac{1}{32}\bar{x}_0+\dfrac{31}{32}x_0=1.140\,63$，已满足条件 $|f(x_1)|<|f(x_0)|$，也即 x_1 已修正了 $\bar{x}_1$ 的严重偏差，使迭代收敛．其计算结果见表 4-4.

表 4-4　　　　　　　　计算结果

k	0	1	2	3	4
λ	1	$\frac{1}{2^5}$	1	1	1
x_k	0.6	1.140 63	1.366 81	1.326 28	1.324 72

一般地说,下山法不再有平方收敛的速度,它的优点在于可能将原来收敛域外的初始值经几次迭代后拉入收敛域内.

4.5　弦截法

用牛顿法解方程 $f(x)=0$,虽然在单根附近具有较快的收敛速度,但它有个明显的缺点,就是需要计算导数 $f'(x)$,当 $f(x)$ 比较复杂时,计算 $f'(x)$ 可能有困难. 但可以设想,若在迭代法中使构造的迭代公式在计算新的近似值 x_{k+1} 时不仅仅用到在点 x_k 上的函数值 $f(x_k)$,而且还用到已算得的点 x_{k-1}, x_{k-2},…等前面几点上的函数值,就有可能提高迭代法的收敛速度.

现设 x_k, x_{k-1} 是 $f(x)=0$ 的近似根,它们对应的函数值为 $f(x_k)$, $f(x_{k-1})$,用差商

$$\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$$

近似代替牛顿迭代公式(4.13)中的导数 $f'(x_k)$,就得到迭代公式

$$x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1}) \qquad (4.16)$$

按迭代公式(4.16)计算方程 $f(x)=0$ 的根 x^* 的近似值的方法就称为弦截法. 此名称是由迭代过程的几何意义得到的. 显然,迭代公式(4.16)可以看作是对方程 $f(x)=0$ 的又一种等价形式

$$x=x-\frac{f(x)}{f(x)-f(x_{k-1})}(x-x_{k-1})=\varphi(x)$$

所建立的迭代公式 $x_{k+1}=\varphi(x_k)$.

弦截法的几何意义如图 4-4 所示.

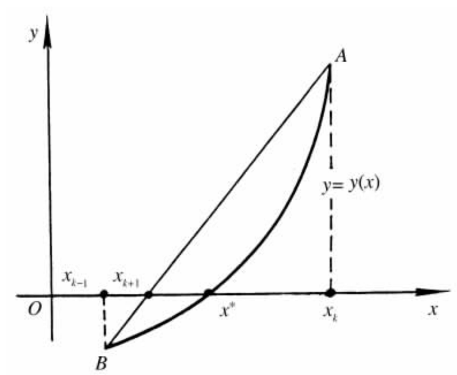


图 4-4

用弦截法求根 x^* 的近似值,在几何上相当于过点 $A(x_k, f(x_k))$,和点 $B(x_{k-1}, f(x_{k-1}))$作直线 AB,其方程为

$$\frac{y-f(x_k)}{x-x_k}=\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$$

然后用弦 AB 与 x 轴的交点的横坐标

$$x_{k+1}=x_k-\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}f(x_k)$$

作为 $f(x)=0$ 的新的近似值.

与牛顿法一样,如果函数 $f(x)$在 $f(x)=0$ 的根 x^* 的附近具有直到二阶的连续导数,且 $f'(x)\neq 0$,则弦截法具有局部收敛性. 即当初始近似值 x_0, x_1 充分接近于 x^* 时,按弦截法迭代公式(4.16)得到的迭代序列$\{x_k\}$收敛于根 x^*. 可以证明弦截法具有超线性收敛速度,且收敛阶为 $P=1.618$.

迭代公式(4.16)与前面介绍的单点迭代法 $x_{k+1}=\varphi(x_k)$有明显的不同,就是在计算 x_{k+1} 时要用到前两步的计算结果 x_k, x_{k-1},所以使用迭代公式(4.16)前必须先给出两个开始值 x_0, x_1,因此,

这种迭代法也称两步法或称双点弦截法．

例 4.7　用弦截法求 $x=e^{-x}$ 在 $x=0.5$ 附近的根．

解　取 $x_0=0.5,x_1=0.6$ 按公式(5.15)迭代的计算结果列于表 4-5.

表 4-5　　计算结果

k	0	1	2	3	4
x_k	0.5	0.6	0.56754	0.567 15	0.567 14

与例 4.3 的结果比较,可以看出弦截法的收敛速度也是较快的.

学习指导

一、基本要求与重点

1. 掌握计算机上常用的一些求非线性方程 $f(x)=0$ 的近似根的数值方法,并能比较各种方法的异同点.

2. 掌握用迭代法求非线性方程近似根的基本思想、几何意义及一些相关的理论和概念,如方法的实用范围、迭代公式、初始值的选取、收敛性的判别及误差估计等.

3. 掌握迭代法的收敛性定理,局部收敛性,收敛阶的概念,并学会使用加速收敛方法,如埃特金(Aitken)加速法.

4. 牛顿法是实用的有效方法,要求熟练常握牛顿迭代公式,并能针对不同的非线性方程构造其牛顿迭代公式及判别其收敛性.

5. 能正确应用所学方法求出给定的非线性方程满足一定精度要求的数值解.

本章重点:一般迭代法,牛顿迭代法,迭代法的收敛性.

二、例题分析与解答

例 1　求方程 $f(x)=\mathrm{e}^{-x}-\sin\left(\dfrac{\pi x}{2}\right)=0$ 在区间$[0,1]$内的实根的近似值，要求误差不超过$\dfrac{1}{2^5}$.

解　用二分法.这里，$a=0,b=1$，且 $f(a)=f(0)=1>0,f(b)=f(1)=-0.6321<0,f(x)=\mathrm{e}^{-x}-\sin\left(\dfrac{\pi x}{2}\right)$在$[0,1]$上连续，所以$[0,1]$是 $f(x)$的有根区间.

用二分法计算结果见表(例)4-1：

表(例)4-1

k	a_k	b_k	x_k	$f(x_k)$的符号
0	0	1	0.5	$f(0.5)-0.1006$
1	0	0.5	0.25	$f(0.25)=0.3961$
2	0.25	0.5	0.375	$f(0.375)=0.1317$
3	0.375	0.5	0.4375	$f(0.4375)=0.0113$

若取 $x_4=\dfrac{1}{2}(0.4375+0.5)=0.46875\approx x^*$，则已满足误差限不超过$\dfrac{1}{2^5}$.

二分法的优点是程序简单，使用十分方便.实际应用中，常用二分法求根的近似值，再用其他方法将根的近似值精确化.

例 2　用迭代法求程 $f(x)=x^3-2x-5=0$ 在区间$[2,3]$上的根，并讨论迭代的敛散性.

解　由于 $f(x)=x^3-2x-5$ 在区间$[2,3]$上连续，且 $f(2)=-1<0,f(3)=16>0$，所以，方程在$[2,3]$上有根.

若把方程改写成下列三种等价的便于迭代的形式：

(1) $x=\sqrt[3]{2x+5}$；

(2) $x=\sqrt{2+\frac{5}{x}}$;

(3) $x=x^3-x-5$.

从而可得出对应的三个迭代格式:

(1) $x_{k+1}=\sqrt[3]{2x_k+5}$;

(2) $x_{k+1}=\sqrt{2+\frac{5}{x_5}}$;

(3) $x_{k+1}=x_k^3-x_k-5$. $(k=0,1,2,\cdots)$

若取相同的初值 $x_0=2$,经分别迭代计算得到的结果列于表(例)4-2 中.

表(例)4-2

	k	0	1	2	3	4	5
迭代(1)	x_k	2	2.080 084	2.092 351	2.094 217	2.094 501	2.094 543
迭代(2)	x_k	2	2.121 320	2.087 348	2.096 517	2.094 017	2.094 697
迭代(3)	x_k	2	1	-5	-125	$-1\,953\,005$	$-7.449\,207\times10^{18}$

从表(例)4-2 中数据的变化趋势看,迭代格式(1)和(2)得到的序列$\{x_k\}$可能是收敛的,而迭代格式(3)可能是发散的.

事实上,由迭代法收敛的充分条件定理 4.1 可知:

(1) 对于迭代格式(1),其迭代函数为 $\varphi(x)=\sqrt[3]{2x+5}$,则$\varphi(x)$在$[2,3]$上具有连续的一阶导数 $\varphi'(x)=\frac{2}{3}(2x+5)^{-\frac{2}{3}}$,且 $\forall x\in[2,3]$,有 $\varphi'(x)>0$,故 $\varphi(x)$单调增加,又 $\varphi(2)=\sqrt[3]{9}>2$,$\varphi(3)=\sqrt[3]{11}<3$,于是,当 $x\in[2,3]$时,$\varphi(x)\in[2,3]$,满足定理 4.1 条件(1).

又,$\varphi'(x)$取正值,且单调递减,所以有

$$\max_{2\leqslant x\leqslant 3}|\varphi'(x)|=\varphi'(2)=\frac{2}{3}(2\times2+5)^{-\frac{2}{3}}<0.154\,081<1,$$

即满足定理 4.1 的条件(2),从而迭代格式(1)收敛.

(2) 对于迭代格式(2),其迭代函数为 $\varphi(x)=\sqrt{2+\dfrac{5}{x}}$,且当 $x\in[2,3]$时,有 $\varphi'(x)=-\dfrac{5}{2x}(2x^2+5x)^{-\frac{1}{2}}<0$,故 $\varphi(x)$是单调减少,但 $\varphi(2)=\sqrt{\dfrac{9}{2}}<3$,$\varphi(3)=\sqrt{\dfrac{11}{3}}<2$,显然不满足定理 4.1 的条件(1),但若在[2,3]的子区间[2,2.5]中考察,则有 $\varphi(2)<2.5$,$\varphi(2.5)=2$,也即满足定理 4.1 的条件(1),又,$\varphi'(x)$在[2,2.5]上取负值且单调递增,从而有 $|\varphi'(x)|=|\varphi'(2)|=\dfrac{5}{4}(2\times 2^2+5\times 2)^{-\frac{1}{2}}<0.294\,628<1$,即满足定理 4.2 的条件(2),从而迭代格式(2)在区间[2,2.5]上收敛.

(3) 对于迭代格式(3),其迭代函数为 $\varphi(x)=x^3-x-5$,在区间[2,3]上有 $\varphi'(x)=3x^2-1$,从而 $\min\limits_{2\leqslant x\leqslant 3}|\varphi'(x)|=\varphi'(2)=11>1$.

在此补充一个判别迭代法发散的充分条件:

若存在 $L\geqslant 1$ 使 $|\varphi'(x)|\geqslant L(\forall x\in[a,b])$,则当 $x_0\neq x_1$ 时,迭代发散.

从而迭代格式(3)当 $x_0\neq x_1$ 时,迭代发散.

例 3 对于给定的正数 c,应用牛顿法于方程 $f(x)=\dfrac{1}{x}-c=0$,并证明:当初始值 x_0 满足条件:$0<x_0<\dfrac{2}{c}$时迭代法收敛.

解 由 $f(x)=\dfrac{1}{x}-c$,有 $f'(x)=-\dfrac{1}{x^2}$,则其牛顿迭代公式为

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}=x_k-\frac{\dfrac{1}{x_k}-c}{-\dfrac{1}{x_k^2}}$$

$$=x_k(2-cx_k),(k=0,1,2,\cdots)$$

这是一个求正数 c 的倒数 $x=\dfrac{1}{c}$ 的一种不用除法计算的迭代法.

下面证明其收敛性:

记第 k 步的迭代误差为 $e_k=x^*-x_k(k=0,1,2,\cdots)$

则 $r_k=ce_k=c(x^*-x_k)=cx^*-cx_k=1-cx_k$,

从而有
$$\begin{aligned} r_{k+1}&=1-cx_{k+1}=1-cx_k(2-cx_k) \\ &=1-cx_k(1+r_k)=r_k-cx_kr_k \\ &=r_k(1-cx_k)=r_k^2, \end{aligned}$$

这是一个关于足标 k 的递推式,反复递推,可得 $r_k=r_0^{2^k}$.

若 $0<x_0<\dfrac{2}{c}$,则 $-1<r_0=1-cx_0<1$,即 $|r_0|<1$,则有 $r_0^{2^k}\to 0$(当 $k\to\infty$ 时),即 $r_k\to 0$,而 c 为常数,从而有迭代误差 $e_k=x^*-x_k\to 0$,也就是当迭代次数 $k\to\infty$ 时,有 $x_k\to x^*$,即迭代法收敛.

例 4 用弦截法求方程 $f(x)=x^3+10x-20=0$ 在区间[1.5,2]内的根的近似值.

解 显然,$f(x)=x^3+10x-20$ 在[1.5,2]上连续,且 $f(1.5)<0$,$f(2)>0$,所以,区间[1.5,2]是 $f(x)=0$ 的有根区间. 若取 $x_0=1.5$,$x_1=2$,用弦截法的计算公式(4.16)计算,结果见表(例)4-3.

表 (例)4-3

k	0	1	2	3	4	5
x_k	1.5	2	1.5844156	1.5934795	1.5945651	1.5945621

计算结果表明,用弦截法迭代 5 次所得的近似解已精确到 8 位有效数字,它的收敛速度也是较快的.

例 5 证明方程 $f(x)=x^3-2x-3=0$ 在区间[1,2]内有唯一的实根 x^*,并问是否能肯定对任意的初始值 $x_0\in[1,2]$,牛顿迭代序列 $\{x_k\}$ 都收敛于 x^*?

证 显然，$f(x)=x^3-2x-3$ 在闭区间$[1,2]$上连续，且 $f(1)f(2)=-4<0$，又 $f'(x)=3x^2-2$ 在区间$[1,2]$上 $f'(x)>0$，由零点定理和函数 $f(x)$ 的单调性知方程 $f(x)=0$ 在$[1,2]$内有唯一的实根.

下面讨论初始值 x_0 的选取. 因为牛顿迭代法对初始值 x_0 的选取要求比较高，x_0 必须充分靠近 x^* 才能保证局部收敛. 在实际应用中，有的实际问题本身可以提供接近于根的初始值，但有的问题却难以确定接近于根的近似值，关于初始值 x_0 的选取除了可用定理 4.3 外，也常用下述定理：

定理 设函数 $f(x)$ 在有限区间$[a,b]$上存在二阶导数，且满足条件：

(1) $f(a)f(b)<0$；

(2) $f'(x)\neq 0, x\in[a,b]$；

(3) $f''(x)$ 在$[a,b]$上不变号；

(4) $$\left|\frac{f(a)}{f'(a)}\right|<b-a, \quad \left|\frac{f(b)}{f'(b)}\right|<b-a.$$

则牛顿迭代法对任意的初始值 $x_0\in[a,b]$ 都收敛于方程 $f(x)=0$ 的唯一根 x^*，且收敛阶数为 2.

本题中由于 $f''(x)=6x>0, x\in[1,2]$，但

$$\left|\frac{f(1)}{f'(1)}\right|=4>2-1=1, \quad \left|\frac{f(2)}{f'(2)}\right|=\frac{1}{10}<2-1=1,$$

因此不能肯定对任意初始值 $x_0\in[1,2]$，牛顿迭代序列 $\{x_k\}$ 都收敛于方程的根 x^*.

但若把区间$[1,2]$缩小为 $\left[\frac{8}{5},2\right]$，则当 $x\in\left[\frac{8}{5},2\right]$ 时，有

$f\left(\frac{8}{5}\right)f(2)<0, f'(x)>0, f''(x)>0$，且

$$\left|\frac{f\left(\frac{8}{5}\right)}{f'\left(\frac{8}{5}\right)}\right|=\frac{263}{710}<0.4=2-\frac{8}{5},$$

$$\left|\frac{f(2)}{f'(2)}\right|=\frac{1}{10}<0.4=2-\frac{8}{5}.$$

则由上述定理知，对任意的初始值 $x_0\in\left[\frac{8}{5},2\right]$，**牛顿迭代序列** $\{x_k\}$ **都收敛于** $f(x)=0$ **的唯一解** x^*.

例 6　设法导出计算 $\frac{1}{\sqrt{a}}$，$(a>0)$ **的牛顿法迭代公式，并要求公式中既无开方运算，又无除法运算.**

解　由于要求迭代式中既无开方运算，又无除法运算，故将计算 $\frac{1}{\sqrt{a}}$，$(a>0)$ **等价化为求** $a-\frac{1}{x^2}=0$ **的正根，**

而此时有

$$f(x)=a-\frac{1}{x^2},f'(x)=\frac{1}{x^3}$$

所以计算 $\frac{1}{\sqrt{a}}$ **的牛顿法迭代公式为**

$$x_{k+1}=x_k-\frac{a-\frac{1}{x_k^2}}{\frac{1}{x_k^3}}=2x_k-ax_k^3$$

$$=(2-ax_k^2)x_k,(k=0,1,2,\cdots)$$

例 7　判断下列非线性方程能否用迭代法求解：

(1) $x=(\cos x+\sin x)/4$；(2) $x=4-2^x$.

解　一个迭代法要具有实用价值，首先要求它是收敛的. 而由收敛性定理 4.1 **知，判断方程** $x=\varphi(x)$ **用迭代法是否收敛，最关键**

的是迭代函数 $\varphi(x)$ 在给定的区间 $[a,b]$ 内是否满足 $|\varphi'(x)|\leqslant L<1$，因此可用此条件来解答本题.

(1) $x=(\cos x+\sin x)/4$ 的迭代函数 $\varphi(x)=(\cos x+\sin x)/4$，则对于 $x\in(-\infty,+\infty)$，都有

$$|\varphi'(x)|=|(-\sin x+\cos x)/4|$$
$$\leqslant\frac{|\sin x|}{4}+\frac{|\cos x|}{4}\leqslant\frac{1}{4}+\frac{1}{4}=\frac{1}{2}<1$$

所以迭代函数 $\varphi(x)$ 满足定理 4.1 条件的(2)，显然也满足定理 4.1 条件的(1)，所以，对任意的初值 x_0，迭代法均收敛，也即可以用迭代法求解.

(2) $x=4-2^x$ 的迭代函数为 $\varphi(x)=4-2^x$，令 $f(x)=x-4+2^x$，则 $f(x)$ 在 $[1,2]$ 上连续，且有 $f(1)f(2)<0$，故区间 $[1,2]$ 为方程 $f(x)=0$ 的有根区间，但

$$|\varphi'(x)|=|-2^x\ln x|\geqslant 2\ln 2\approx 1.38629>1$$

故不能用迭代公式 $x_{k+1}=4-2^{x_k}$ 来迭代.

若把原方程改写为 $x=\ln(4-x)/\ln 2$，此时迭代函数 $\varphi(x)=\ln(4-x)/\ln 2$，则有

$$|\varphi'(x)|=\left|\frac{-1}{4-x}\cdot\frac{1}{\ln 2}\right|<\frac{1}{4-2}\cdot\frac{1}{\ln 2}=\frac{1}{2\ln 2}<1,$$

满足定理 4.1，故可用迭代公式 $x_{k+1}=\ln(4-x_k)/\ln 2$ 来求解.

习　题　四

1. 用二分法求方程 $f(x)=\sin x-\left(\frac{x}{2}\right)^2=0$ 在区间 $[1.5,2]$ 内的实根的近似值，并指出其误差.

2. 方程 $x^3-x^2-1=0$ 在 $x_0=1.5$ 附近有根，把方程写成三种不同的等价形式，并建立对应的迭代公式如下：

（1）$x=1+\frac{1}{x^2}$， **迭代公式：** $x_{k+1}=1+\frac{1}{x_k^2}$；

（2）$x^3=1+x^2$， **迭代公式：** $x_{k+1}=\sqrt[3]{1+x_k^2}$；

（3）$x^2=\frac{1}{x-1}$， **迭代公式：** $x_{k+1}=\frac{1}{\sqrt{x_k-1}}$.

试判断各种迭代格式在 $x_0=1.5$ **附近的收敛性，并估计收敛速度. 选一种收敛格式，计算出具有四位有效数字的近似根.**

3. **用牛顿法求** $f(x)=x-\cos x=0$ **在** $x_0=1$ **附近的实根，要求满足精度** $|x_{k+1}-x_k|<0.001$.

4. **应用牛顿法解方程** $x^3-a=0$，**导出求立方根** $\sqrt[3]{a}$ **的近似公式.**

5. **用弦截法求方程** $x^3-3x-1=0$ **在** $x_0=2$ **附近的实根，设取** $x_0=1.9$，**计算到四位有效数字为止.**

第五章 线性代数方程组的数值解法

在科学研究和工程计算中，经常会遇到求解线性代数方程组的问题，且许多有效的数值计算方法中，作为关键的一步就需要求解线性代数方程组，例如，网络问题、非线性问题线性化、样条插值、曲线拟合、微分方程离散化后的求解问题等等，都导致求解线性代数方程组．因此，线性代数方程组的数值解法是数值计算的重要内容．

本章讨论 n 阶线性代数方程组

$$\begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=b_n \end{cases} \tag{5.1}$$

的数值解法，其中 $x_i(i=1,2,\cdots,n)$ 为未知元，$a_{ij}(i,j=1,2,\cdots,n)$，$b_i(i=1,2,\cdots,n)$ 为常数．

或写成矩阵形式，

$$\boldsymbol{AX}=\boldsymbol{b}$$

式中，$\boldsymbol{A}=(a_{ij})_{n\times n}$ 是由方程组的系数组成的 n 阶矩阵，称为系数矩阵；$\boldsymbol{b}=(b_1,b_2,\cdots,b_n)^{\mathrm{T}}$ 是由右端项组成的 n 维列向量，称为右端向量；$\boldsymbol{X}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 是由未知元组成的 n 维列向量，称为解向量．

由线性代数知识知，当且仅当 $\det(\boldsymbol{A})\neq 0$，即 $\boldsymbol{A}$ 非奇异时，根据克莱姆法则，线性代数方程组(5.1)存在唯一解．并可表示为两个行列式之比

$$x_i=\frac{D_i}{D},(i=1,2,\cdots,n)$$

其中,$D=|\boldsymbol{A}|$,D_i 是将行列式 D 的第 i 列用右端向量 $\boldsymbol{b}$ 来代替所成的行列式,但这种方法在实际使用时仅适用低阶方程组,对于高阶方程组不能用,这可从工作量来分析.

用克莱姆法则求解一个 n 阶方程组,首先,要计算 $n+1$ 个 n 阶行列式的值,而按定义把行列式展开进行计算,工作量就会大得惊人. n 阶行列式有 $n!$ 项,每一项含 n 个因子,所以计算一个 n 阶行列式需要做 $(n-1)(n!)$ 个乘法,则计算一个 n 阶方程组就要做 $(n+1)(n-1)(n!)$ 次乘法,还要做 n 个除法. 若只算乘法的工作量,求解一个 20 阶线性代数方程组,其工作量是

$$20!\times19\times21\approx9.7\times10^{20}\text{ 次乘法}$$

若用计算速度为每秒一亿次的计算机也要算 30 多万年. 因此克莱姆法则尽管在理论上是非常完善的,但在实际计算时,仅就上述如此大的计算量来说,也是难以实现的. 因此学习和研究依靠计算机求解线性代数方程组的行之有效的数值解法在数值计算中占有十分重要的地位.

目前,计算机上解线性代数方程组的数值方法尽管很多,但归纳起来,大致可以分为两大类:一类是直接法(也称精确解法),另一类是迭代法.

所谓直接法是指这样一类方法:若不计运算过程的舍入误差(即假定运算都是精确进行的),经过有限次运算后,能求得方程组的精确解的方法.

但实际计算时,由于机器只能用有限位小数作运算,所以不可能没有舍入误差. 由于舍入误差的存在和影响,这种方法也只能求得方程组的近似解.

本章将介绍这类方法中最基本的高斯(Gauss)消去法和矩阵分解法等,由于直接法的准确性和可靠性,对于以中低阶($n<100$)的稠密矩阵为系数矩阵的相应方程组,直接法是经常使用的有效

方法. 近些年来，直接法在求解较大型的带型稀疏矩阵方程组方面也取得了较大进展.

所谓迭代法就是用通过某种极限过程去逐步逼近方程组精确解的方法. 由于迭代法不需要存贮系数矩阵的零元素，因而迭代法适用于求解系数矩阵为大型稀疏矩阵的方程组，同时，还由于计算机程序设计比较简单，原始系数矩阵在计算中始终保持不变，所以迭代法被广泛使用，但迭代法存在收敛性与收敛速度的问题，往往要求方程组的系数矩阵具有某些特殊的性质. 另外，用迭代法时，我们绝不可能把极限过程进行到底，而只能把迭代进行有限多次，使迭代若干次所得的结果达到一定的精确度. 因而在有限步内我们所得到的解也只能是近似解.

本章将介绍雅可比(Jacobi)迭代法，赛德尔(Seidel)迭代法和超松弛迭代法.

下面先介绍求解线代数方程组的直接法中几种常用的数值方法，并均假设方程组(5.1)的系数矩阵非奇异.

5.1 高斯(Gauss)消去法

高斯消去法是一个古老的求解线性代数方程组的直接法，但由它的改进，变形得到的如选主元消去法、三角分解法等，仍然是目前计算机上解低阶稠密矩阵方程组的常用有效方法.

高斯消去法是建立在逐次消去未知元的基础上，它的基本做法是把方程组(5.1)用逐次消去未知元的方法化为与其等价的(或者说是具有相同解的)三角形方程组

$$\begin{cases} u_{11}x_1+u_{12}x_2+\cdots+u_{1n}x_n=b_1 \\ \qquad\qquad u_{22}x_2+\cdots+u_{2n}x_n=b_2 \\ \qquad\qquad\qquad\cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad\qquad\qquad\qquad u_{nn}x_n=b_n \end{cases} \tag{5.2}$$

这个过程称为消元过程,得到(5.2)式以后,求解就很容易了,只要从最后一个方程依次往上代,就可以逐个求出未知元 $x_n, x_{n-1}, \cdots, x_1$,我们称这个计算过程为回代过程. 这就是高斯消去法.

由线性代数知识知道,不论是消元过程还是回代过程都不需要对未知元或方程作真正的运算,只要对它们的系数和右端项作运算就足够了,换句话说,只要把方程组的系数和右端项从方程中分离出来,那么,消去法完全可通过增广矩阵的行的初等变换来实现,其消去过程的基本步骤示意如下:

$$[\boldsymbol{A} \vdots \boldsymbol{b}] =$$

$$\begin{bmatrix} \times & \times & \cdots & \times & \vdots & \times \\ \times & \times & \cdots & \times & \vdots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \times & \cdots & \times & \vdots & \times \end{bmatrix} \xrightarrow{(1)\text{步}} \begin{bmatrix} \times & \times & \cdots & \times & \vdots & \times \\ 0 & \times & \cdots & \times & \vdots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \cdots & \times & \vdots & \times \end{bmatrix}$$

$$\xrightarrow{(2)\text{步}} \begin{bmatrix} \times & \times & \times & \cdots & \times & \vdots & \times \\ 0 & \times & \times & \cdots & \times & \vdots & \times \\ 0 & 0 & \times & \cdots & \times & \vdots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \times & \cdots & \times & \vdots & \times \end{bmatrix} \longrightarrow \cdots \xrightarrow{(n-1\text{ 步})}$$

$$\begin{bmatrix} \times & \cdots & \cdots & \cdots & \times & \vdots & \times \\ & \ddots & & & \vdots & \vdots & \vdots \\ & & \ddots & & \vdots & \vdots & \vdots \\ & & & \ddots & \vdots & \vdots & \vdots \\ & & & & \ddots & \vdots & \vdots \\ & & & & \times & \vdots & \times \end{bmatrix}$$

然后自下而上进行回代,按上述规定的步骤做,既便于编程序又节约计算工作量. 具体的计算方法常分为顺序高斯消去法、列主元素

高斯消去法和全主元素高斯消去法.

一、顺序高斯消去法

求解 n 阶线代数方程组(5.1)的顺序高斯消去法的步骤如下：

1. 消元过程

记系数矩阵 $\boldsymbol{A}=\boldsymbol{A}^{(1)}=(a_{ij}^{(1)})_{\boldsymbol{n}\times\boldsymbol{n}}$，

右端向量 $\boldsymbol{b}=\boldsymbol{b}^{(1)}=(b_1,b_2,\cdots,b_n)^{\mathrm{T}}$

(1) 第一步：设 $a_{11}^{(1)}\neq 0$，记 $l_{i1}=a_{i1}^{(1)}/a_{11}^{(1)}(i=2,3,\cdots,n)$，然后将增广矩阵$[A^{(1)}\vdots b^{(1)}]$中第 i 行减去第 1 行乘以 l_{i1}，$(i=2,3,\cdots,n)$，完成第一次消元，得

$$[\boldsymbol{A}^{(2)}\vdots\boldsymbol{b}^{(2)}]=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & \vdots & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & \vdots & b_2^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & \vdots & b_n^{(2)} \end{bmatrix} \tag{5.3}$$

其中，

$a_{ij}^{(2)}=a_{ij}^{(1)}-l_{i1}a_{ij}^{(1)}$，$(i,j=2,3,\cdots,n)$，$b_i^{(2)}=b_i^{(1)}-l_{i1}b_1^{(1)}$，$(i=2,3,\cdots,n)$，

这样，就得到了与原方程组 $\boldsymbol{A}^{(1)}\boldsymbol{X}=\boldsymbol{b}^{(1)}$ 同解的方程组 $\boldsymbol{A}^{(2)}\boldsymbol{X}=\boldsymbol{b}^{(2)}$.

(2) 第二步：设 $a_{22}^{(2)}\neq 0$，记 $l_{i2}=a_{i2}^{(2)}/a_{22}^{(2)}$，$(i=3,4,\cdots,n)$，然后将矩阵$[\boldsymbol{A}^{(2)}\vdots\boldsymbol{b}^{(2)}]$中第 i 行减去第 2 行乘以 $l_{i2}(i=3,4,\cdots,n)$，完成第二次消元，得

$$[\boldsymbol{A}^{(3)}\vdots\boldsymbol{b}^{(3)}]=\left[\begin{array}{cccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} & & b_n^{(3)} \end{array}\right] \tag{5.4}$$

其中，

$$a_{ij}^{(3)}=a_{ij}^{(2)}-l_{i2}a_{2j}^{(3)},(i,j=3,4,\cdots,n),$$

$$b_i^{(3)}=b_i^{(2)}-l_{i2}b_2^{(2)},(i=3,4,\cdots,n),$$

这样，就得到了与原方程组 $\boldsymbol{A}^{(1)}\boldsymbol{X}=\boldsymbol{b}^{(1)}$ 同解的方程组 $\boldsymbol{A}^{(3)}\boldsymbol{X}=\boldsymbol{b}^{(3)}$.

(3) 第 k 步：重复上述过程，在完成 $k-1$ 消元后，就获得与原方程组 $\boldsymbol{A}^{(1)}\boldsymbol{X}=\boldsymbol{b}^{(1)}$ 同解的方程组 $\boldsymbol{A}^{(k)}\boldsymbol{X}=\boldsymbol{b}^{(k)}$. 则第 k 次消元为：设 $a_{kk}^{(k)}\neq0$，记 $l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)},(i=k+1,\cdots,n)$，然后将矩阵中第 i 行减去第 k 行乘以 $l_{ik},(i=k+1,\cdots,n)$，完成第 k 次消元，得

$$[\boldsymbol{A}^{(k+1)}\vdots\boldsymbol{b}^{(k+1)}]=\begin{bmatrix} a_{11}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} & \vdots \\ & \ddots & & & \vdots & \vdots \\ & & a_{kk}^{(k)} & \cdots & \cdots & a_{kn}^{(k)} \vdots & b_{kn}^{(k)} \\ & & & a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} & \vdots \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} & \vdots \end{bmatrix} \tag{5.5}$$

其中 $$a_{ij}^{(k+1)}=a_{ij}^{(k)}-l_{ik}a_{kj}^{(k)},(i,j=k+1,\cdots,n),$$

$$b_i^{(k+1)}=b_i^{(k)}-l_{ik}b_k^{(k)},(i=k+1,\cdots,n).$$

这样，就得到了与原方程组同解的方程组 $\boldsymbol{A}^{(k+1)}\boldsymbol{X}=\boldsymbol{b}^{(k+1)}$. 如此继续下去，完成 $n-1$ 次消元后，原方程组(5.1)化成同解的上三角形方程组 $\boldsymbol{A}^{(n)}\boldsymbol{X}=\boldsymbol{b}^{(n)}$，即

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots\cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots\cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}=\begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(n)} \end{bmatrix} \tag{5.6}$$

2. 回代过程

由方程组(5.6)自下而上按未知元 x_i 的足标逆序逐步回代得原方程组(5.1)的解.

称 $a_{kk}^{(k)}$ $(k=1,2,\cdots,n)$ 为各步消元的主元素,称 l_{ik} $\left(\begin{matrix}k=1,2,\cdots,n-1\\ i=k+1,\cdots,n\end{matrix}\right)$ 为各步消元的乘数,若用 k 表示消元过程的序号,则顺序高斯消去法的计算公式如下:

(1) 消元过程(设 $a_{kk}^{(k)}\neq 0$)

对 $k=1,2,\cdots,(n-1)$ 依次计算:

$$\begin{cases}l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}\\ a_{ij}^{(k+1)}=a_{ij}^{(k)}-l_{ik}a_{kj}^{(k)}\\ b_i^{(k+1)}=b_i^{(k)}-l_{ik}b_k^{(k)}\end{cases},(i,j=k+1,k+2,\cdots,n) \tag{5.7}$$

(2) 回代过程

$$\begin{cases}x_n=b_n^{(n)}/a_{nn}^{(n)}\\ x_k=(b_k^{(k)}-\sum\limits_{j=k+1}^{n}a_{kj}^{(k)}x_j)/a_{kk}^{(k)},\ (k=n-1,n-2,\cdots,2,1)\end{cases} \tag{5.8}$$

在顺序高斯消去法的消元过程中,由公式(5.7)知,乘法运算次数为

$$\sum_{K=1}^{n-1}(n-k)(n-k)+\sum_{k=1}^{n-1}(n-k)=\frac{1}{3}(n^3-n)$$

除法运算次数为

$$\sum_{k=1}^{n-1}(n-k)=\frac{1}{2}n(n-1)$$

在回代过程中,由公式(5.8)知,乘法运算次数为

$$\sum_{k=1}^{n-1}(n-k)=\frac{1}{2}n(n-1)$$

除法运算次数为 n，所以用顺序高斯消去法求解一个 n 阶线性代数方程组(5.1)所需的乘除法运算次数总计为

$$\frac{1}{3}(n^3-n)+\frac{1}{2}n(n-1)+\frac{1}{2}n(n-1)+n=\frac{n^3}{3}+n^2-\frac{n}{3}$$

与克莱姆法则相比，计算工作量小得多. 例如，当 $n=20$ 时，用顺序高斯消去法只需 3060 次乘除法运算.

二、列主元素高斯消去法

上面介绍的顺序高斯消去法是按方程组的未知元给定的自然顺序逐个消元的，显然，消去法能够进行到底的条件是各步消元的主元素 $a_{kk}^{(k)}$ 全不能为零，如果在消元过程中发现某个主元素为零，即使 $\det(\boldsymbol{A})\neq 0$，则消元过程将无法进行. 其次，即使主元素不为零，但如果主元素 $a_{kk}^{(k)}$ 的绝对值很小，用它作除数，将使该步消元的乘数绝对值很大，势必造成舍入误差的严重扩散，以至于方程组解的精确度受到严重的影响. 下面用一个数值解的简例来说明主元选取的重要性.

例 5.1 求解二元线性方程组

$$\begin{cases}\dfrac{1}{10\,000}x_1+x_2=1\\ x_1+x_2=2\end{cases}$$

可用克莱姆法则算得其精确解为

$$x_1=\frac{10\,000}{9\,999},\qquad x_2=\frac{9\,998}{9\,999}$$

现在如果用三位浮点十进制数求解：

(1) 用顺序高斯消去法求解

消元过程的矩阵可表示为

$$\left[\begin{array}{cc:c}0.000\,100 & 1.00 & 1.00\\ 1.00 & 1.00 & 2.00\end{array}\right]\xrightarrow{\text{第一次消元}}$$

$$\begin{bmatrix} 0.000100 & 1.00 & \vdots & 1.00 \\ 0.00 & -10000 & \vdots & -10000 \end{bmatrix}$$

其中，$a_{11}^{(1)}=0.000100$，$l_{21}=a_{21}^{(1)}/a_{11}^{(1)}=\dfrac{1.00}{0.000100}=1000.$

回代求解得　$x_2=1.00$，$x_1=0.00$

显然，$x_1=0.00$ 不是原方程组的解，其根本原因就是在第一步消元时，取主元 $a_{11}^{(1)}=0.000100$ 太小造成的，在第一次消元中计算乘数 l_{21} 时，用绝对值相对地太小的数作除数，从而带入了大的舍入误差，再经传播，使误差被扩大了差不多 10000 倍，从而近似解 x_1 完全失去了近似意义. 然后，如果交换原方程组的两个方程的次序，还用三位浮点十进制数运算.

(2) 用行序变换后消元求解

则有

$$\begin{bmatrix} 1.00 & 1.00 & \vdots & 2.00 \\ 0.000100 & 1.00 & \vdots & 1.00 \end{bmatrix} \xrightarrow{\text{第一次消元}} \begin{bmatrix} 1.00 & 1.00 & \vdots & 2.00 \\ 0.00 & 1.00 & \vdots & 1.00 \end{bmatrix}$$

回代求解得　$x_2=1.00$，$x_1=1.00$

它与方程组的精确解已相当接近了，是较好的近似解. 其根本原因是消元过程中避免了用“小主元”.

上例充分说明了，在高斯消去法的消元过程中选主元的必要性，以及选主元技巧的有效性. 下面就来介绍一种带有选主元技巧的消去法——列主元素高斯消去法.

对方程组 $\boldsymbol{AX}=\boldsymbol{b}$，仍按 $x_1, x_2, \cdots, x_n$ 的顺序依次进行高斯消元，只是在每一步消元前增加一步按列选主元的工作，即

第一步：在消元前先在系数矩阵 $\boldsymbol{A}$ 的第 1 列元素中选取绝对值最大的元素为主元素.

即

$$[\boldsymbol{A} \vdots \boldsymbol{b}]=\left[\begin{array}{ccccc:c} \boxed{a_{11}} & a_{12} & \cdots & a_{1n} & & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & & b_2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & & b_n \end{array}\right]$$

设取到$|a_{i_1,1}|=\max\limits_{1\leqslant i\leqslant n}|a_{i1}|$,若$|a_{i_1,1}|=0$,则第1列全为0,即系数矩阵$A$为奇异的,这与假设$A$为非奇异矛盾,所以$|a_{i_1,1}|\neq 0$;

若$a_{i_1,1}$不在(1,1)位置上,则可通过交换第一行与第i_1行,把$a_{i_1,1}$换到(1,1)位置作为主元素,记行交换后的增广矩阵为$[\boldsymbol{A}^{(1)} \vdots \boldsymbol{b}^{(1)}]$,然后再进行高斯消去法的第一次消元,得矩阵$[\boldsymbol{A}^{(2)} \vdots \boldsymbol{b}^{(2)}]$.

第二步:在矩阵$\boldsymbol{A}^{(2)}$的第2列元素$a_{i2}^{(2)}(i=2,3,\cdots,n)$中选取绝对值最大的元素为第二步主元素,设$|a_{i_2,2}^{(2)}|=\max\limits_{2\leqslant i\leqslant n}|a_{i2}^{(2)}|$,显然$|a_{i_2,2}^{(2)}|\neq 0$,否则与$A$非奇异矛盾.若$a_{i_2,2}^{(2)}$不在(2,2)位置,则将增广矩阵$[\boldsymbol{A}^{(2)} \vdots \boldsymbol{b}^{(2)}]$的第2行与第$i_2$行交换,把$a_{i_2,2}^{(2)}$换到(2,2)位置,作为主元素,再进行高斯消去法的第二次消元,得矩阵$[\boldsymbol{A}^{(3)} \vdots \boldsymbol{b}^{(3)}]$.

如此进行下去,假设已经实现了前$k-1$次消元$(1\leqslant k\leqslant n)$,得到矩阵$[\boldsymbol{A}^{(k)} \vdots \boldsymbol{b}^{(k)}]$.

第k步:在矩阵$\boldsymbol{A}^{(k)}$的第k列元素$a_{ik}^{(k)}(i=k,k+1,\cdots,n)$中选取绝对值最大的元素为第$k$步主元素,

即

$$[\boldsymbol{A}^{(k)} \vdots \boldsymbol{b}^{(k)}]=\left[\begin{array}{cccccc:c} a_{11}^{(1)} & a_{12}^{(1)}\cdots & \cdots & \cdots & & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)}\cdots & \cdots & \cdots & & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & & & \vdots & \vdots \\ & & & \boxed{a_{kk}^{(k)}} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\ & & & \vdots & \ddots & \vdots & \vdots \\ & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)} \end{array}\right]$$

设$|a_{i_k,k}^{(k)}|=\max\limits_{k\leqslant i\leqslant n}|a_{ik}^{(k)}|$，若$|a_{i_k,k}^{(k)}|=0$，这与$A$非奇异矛盾. 若$a_{i_k,k}^{(k)}$不在$(k,k)$位置，则将增广矩阵$[\boldsymbol{A}^{(k)} \vdots \boldsymbol{b}^{(k)}]$的第$k$行与第$i_k$行交换，把$a_{i_k,k}^{(k)}$换到$(k,k)$位置，作为第$k$步主元素，然后再进行高斯消去法的第$k$次消元.

如此继续，在每步消元前都要按列选取主元素，直到完成$n-1$次消元为止，此时，原方程组已被化成等价的上三角形方程组，然后用高斯消去法的回代过程，求得方程组的全部解. 上述选主元素的方法，由于每步是按列选主元素，故称为列主元素高斯消去法. 例 5.1 的第二种解法用的就是列主元素高斯消去法.

此外，还有所谓全主元素高斯消去法，即在进行k步消元前，在$\boldsymbol{A}^{(k)}$的位于右下角的$n-(k-1)$阶子矩阵

$$\begin{bmatrix} a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & \ddots & \vdots \\ a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{bmatrix}$$

中选取绝对值最大的元素作为主元素，再进行行和列的交换，把选到的主元素调至(k,k)位置，再按高斯消去过程进行消元. 初看起来全主元素消去法比列主元素消去法更好，精度更高，但全主元素法在选主元时要花费大量机器的时间进行元素的比较，计算工作量比列主元素法大得多，而且在全主元素法中，随着换列变换，未知元的排列顺序也要作相应的变换，因而程序较复杂. 事实上，列主元素高斯消去法的精度虽然稍低于全主元素法，但其计算简单，工作量小，且可以证明，它与全主元素法同样具有良好的数值稳定性，因此到目前为止，列主元素高斯消去法仍然是计算机上解系数矩阵为中小型稠密矩阵的线性代数方程组的最实用、最有效的方法之一.

5.2 三角分解法

把一个 n 阶矩阵 A 分解成结构简单的三角形矩阵的乘积称为矩阵的三角分解. 本节我们将进一步用矩阵的理论来分析高斯消去法,从而导出另一种求解线性代数方程组的直接法——三角分解法.

下面的讨论将要给出一个带关键性的问题,高斯消去法在本质上就是将系数方阵 $\boldsymbol{A}$ 分解为两个三角阵的乘积,即 $\boldsymbol{A}=\boldsymbol{LU}$,其中,$\boldsymbol{L}$ 是单位下三角阵,$\boldsymbol{U}$ 是上三角阵,所以三角分解法也可以说是消去法的一种变形.

一、高斯消去法与矩阵的三角分解

下面我们对一般的 n 阶方程组 $\boldsymbol{AX}=\boldsymbol{b}$,从矩阵变换和运算角度,对高斯消去法作进一步的分析.

从 5.1 节讨论可知,对方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 用顺序高斯消去法求解的每一步消元实际上相当于对原方程组的系数矩阵作一个行的初等变换. 例如,第一步从第 2 个到第 n 个方程中消去未知元 x_1,得到等价的新方程组的系数矩阵为

$$\boldsymbol{A}^{(2)}=\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix},\text{右端项 } \boldsymbol{b}^{(2)}=\begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{bmatrix}$$

从矩阵运算角度来分析,以上变化相当于做这样一件事:用 $l_{i1}=a_{i1}^{(1)}/a_{11}^{(1)}$,$(i=2,3,\cdots,n)$构造一个"行初等变换方阵".

$$
\boldsymbol{L}_1 = \begin{bmatrix} 1 & & & & \\ -l_{21} & 1 & & & \\ -l_{31} & 0 & & 1 & \\ \vdots & \vdots & \ddots & & \ddots \\ -l_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix}
$$

然后,用 $\boldsymbol{L}_1$ 去左乘系数矩阵

$$
\boldsymbol{A}^{(1)} = (a_{ij}^{(1)})_{n\times n}
$$

和右端向量

$$
\boldsymbol{b}^{(1)} = (b_1^{(1)}, \cdots, b_n^{(1)})^{\mathrm{T}},
$$

即得到与高斯消去法第一步同样的结果:$\boldsymbol{A}^{(1)} \to \boldsymbol{A}^{(2)}, \boldsymbol{b}^{(1)} \to \boldsymbol{b}^{(2)}$.

也即有 $$\boldsymbol{L}_1 \boldsymbol{A}^{(1)} = \boldsymbol{A}^{(2)}, \boldsymbol{L}_1 \boldsymbol{b}^{(1)} = \boldsymbol{b}^{(2)}$$

第二步消去过程 $\boldsymbol{A}^{(2)} \to \boldsymbol{A}^{(3)}, \boldsymbol{b}^{(2)} \to \boldsymbol{b}^{(3)}$,相当于用初等行变换矩阵

$$
\boldsymbol{L}_2 = \begin{bmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & -l_{32} & 1 & & & \\ & & 0 & & \ddots & \\ \vdots & \vdots & \vdots & \ddots & & \\ 0 & -l_{n2} & 0 & \cdots & 0 & 1 \end{bmatrix}
$$

分别去左乘 $\boldsymbol{A}^{(2)}, \boldsymbol{b}^{(2)}$,即

$$
\boldsymbol{L}_2 \boldsymbol{A}^{(2)} = \boldsymbol{A}^{(3)}, \boldsymbol{L}_2 \boldsymbol{b}^{(2)} = \boldsymbol{b}^{(3)},
$$

其中,$l_{i2} = a_{i2}^{(2)} / a_{22}^{(2)}, (i = 3, 4, \cdots, n)$

一般地,第 k 步消去过程是把 $\boldsymbol{A}^{(k)} \to \boldsymbol{A}^{(k+1)}, \boldsymbol{b}^{(k)} \to \boldsymbol{b}^{(k+1)}$,

这相当于用初等行变换矩阵

$$
\boldsymbol{L}_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{n,k} & \cdots & & 1 \end{bmatrix}
$$

分别去左乘第$(k-1)$次消元后得到的系数矩阵和右端向量.

即
$$
\boldsymbol{L}_k\boldsymbol{A}^{(k)}=\boldsymbol{A}^{(k+1)},\boldsymbol{L}_k\boldsymbol{b}^{(k)}=\boldsymbol{b}^{(k+1)}
$$

依次下去,直到最后的$(n-1)$步消元,即用初等行变换矩阵

$$
\boldsymbol{L}_{n-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -l_{n,n-1} & 1 \end{bmatrix}
$$

分别去左乘 $\boldsymbol{A}^{(n-1)}$ 和 $\boldsymbol{b}^{(n-1)}$,得

$$
\boldsymbol{L}_{n-1}\boldsymbol{A}^{(n-1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix} = \boldsymbol{A}^{(n)}
$$

$$
\boldsymbol{L}_{n-1}\boldsymbol{b}^{(n-1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(n)} \end{bmatrix} = \boldsymbol{b}^{(n)}
$$

这里的 $\boldsymbol{A}^{(n)}$ 就是高斯消去法的消去过程最终得到的等价方程组的上三角状系数方阵. 为了下面讨论方便,记

$$
\boldsymbol{A}^{(n)}=\boldsymbol{U},\quad \boldsymbol{b}^{(n)}=\boldsymbol{Y}
$$

从上面的分析看到，高斯消去法每消去一个未知元相当于系数矩阵和右端向量左乘一个相应的初等行变换矩阵 $\boldsymbol{L}_k$，所以经过 $n-1$ 步后得到的上三角阵 $\boldsymbol{U}$ 和向量 $\boldsymbol{Y}$ 与原方程组的系数矩阵 $\boldsymbol{A}$ 和右端向量 $\boldsymbol{b}$ 的关系为

$$\boldsymbol{U}=\boldsymbol{L}_{n-1}\boldsymbol{L}_{n-2}\cdots\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A}$$

$$\boldsymbol{Y}=\boldsymbol{L}_{n-1}\boldsymbol{L}_{n-2}\cdots\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{b}$$

其中，矩阵 $\boldsymbol{L}_k$ 是一种特殊的单位下三角方阵：

$$\boldsymbol{L}_k=\begin{bmatrix}1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & -l_{k+2,k} & & \ddots & \\ & & \vdots & & & \\ & & -l_{nk} & & & 1\end{bmatrix},(k=1,2,\cdots,n-1)$$

称为 Frobenius 矩阵，除了它的第 k 列对角线以下元素为

$$-l_{ik}=-a_{ik}^{(k)}/a_{kk}^{(k)},(i=k+1,k+2,\cdots,n)$$

外，其他元素与单位矩阵 $\boldsymbol{I}$ 完全相同. 用代数知识很容易证明它具有下列性质：

(1) $\boldsymbol{L}_k$ 的逆阵存在且为

$$\boldsymbol{L}_k^{-1}=\begin{bmatrix}1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & l_{k+1,k} & 1 & \\ & & \vdots & & \ddots & \\ & & l_{n,k} & & & 1\end{bmatrix}$$

即 $\boldsymbol{L}_k^{-1}$ 仍是一个单位下三角阵，且正好可以看作是把 $\boldsymbol{L}_k$ 的非对角元素改变一个符号得到.

(2)
$$\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}=\begin{bmatrix}1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & & & 1\end{bmatrix}$$

且一般有

$$\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}=\begin{bmatrix}1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1\end{bmatrix}\overset{\text{记}}{=\!=}\boldsymbol{L}$$

显然，$\boldsymbol{L}$ 是对角元素皆等于 1 的一个下三角形方阵，而它的非对角元 l_{ik} 正好是高斯消去法中的乘数

$$l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)},(i=k+1,k+2,\cdots,n)$$

到此，我们由

$$\boldsymbol{L}_{n-1}\boldsymbol{L}_{n-2}\cdots\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A}=\boldsymbol{U}$$

和
$$\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}=\boldsymbol{L}$$

已导出 $\boldsymbol{A}$ 的分解式

$$\boldsymbol{A}=\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}\boldsymbol{U}=\boldsymbol{L}\boldsymbol{U}$$

式中，$\boldsymbol{L}$ 是单位下三角阵，$\boldsymbol{U}$ 是上三角阵.

从而我们推导出了在高斯消去法的讨论中起关键作用的结果. 顺序高斯消去法的消去过程实质上就是用一连串初等行变换方阵 $\boldsymbol{L}_k$ 相继左乘 $\boldsymbol{A}$ 与 $\boldsymbol{b}$ 最终得到 $\boldsymbol{U}$ 和 $\boldsymbol{Y}$.

公式 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$ 叫做方阵 $\boldsymbol{A}$ 的三角状分解或 $\boldsymbol{L}\boldsymbol{U}$ 分解.当然,这种分解是要有条件的,它要求在消元过程中出现的主元素 $a_{11}^{(1)}$,$a_{22}^{(2)},\cdots,a_{kk}^{(k)},\cdots$ 均不为零(因 $l_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}$,分母显然不能为零).

那么 $\boldsymbol{A}$ 满足什么条件时,才能保证 $a_{kk}^{(k)}\neq 0,(k=1,2,\cdots,n)$ 呢?

事实上,由顺序高斯消去法的消元过程可以看出,矩阵 $\boldsymbol{A}=\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(n)}$ 的各阶顺序主子式的值都保持不变,即

$$\Delta_k=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{vmatrix}=a_{11}^{(1)}a_{22}^{(2)}\cdots a_{kk}^{(k)},(k=1,2,\cdots,n)$$

因此,当 $a_{kk}^{(k)}\neq 0,(k=1,2,\cdots,n)$ 时,必有 $\Delta_k\neq 0,(k=1,2,\cdots,n)$;反之,可用数学归纳法证明,当 $\Delta_k\neq 0$ 时必有 $a_{kk}^{(k)}\neq 0(k=1,2,\cdots,n)$,也即有下述定理:

定理 5.1 用顺序高斯消去法求解方程组 $\boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}$ 时的主元素 $a_{kk}^{(k)}\neq 0,(k=1,2,\cdots,n)$ 的充要条件是 n 阶矩阵 $\boldsymbol{A}$ 的所有顺序主子式均不为零,即 $a_{11}\neq 0,\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}\neq 0,\cdots,\det(\boldsymbol{A})\neq 0.$

从而,由上面讨论得到的求解线代数方程组 $\boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}$ 的顺序高斯消去法的实质是将系数矩阵 $\boldsymbol{A}$ 分解为两个三角阵的乘积,所以可得到矩阵的三角分解定理:

定理 5.2 设 $\boldsymbol{A}\in\boldsymbol{R}^{n\times n}$,如果 $\boldsymbol{A}$ 的顺序主子式 $\Delta_k\neq 0,(k=1,2,\cdots n)$,则 $\boldsymbol{A}$ 存在唯一的分解式

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}.$$

其中,$\boldsymbol{L}$ 为 n 阶单位下三角阵,$\boldsymbol{U}$ 为 n 阶上三角阵.

证明 在定理的假设条件下,由上述高斯消去法的矩阵分析,$\boldsymbol{A}$ 的三角分解式:$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$ 的存在性已经证明,下面再证明 $\boldsymbol{A}$ 的三

角分解的唯一性.

不妨设矩阵 A 有两种三角分解:

$$A = L_1U_1 = LU \tag{5.9}$$

其中, L_1, L 为单位下三角阵, U_1, U 为上三角阵.

由假设 A 非奇异,则 L_1, L, U_1, U 均为非奇异阵,于是由(5.9)式有

$$L^{-1}L_1 = UU_1^{-1} \tag{5.10}$$

由于单位下三角阵 L 的逆阵 L^{-1} 仍是单位下三角阵,上三角阵 U_1 的逆阵 U_1^{-1} 仍是上三角阵,则(5.10)式的左边为单位下三角阵,而右边为上三角阵,所以

$$L^{-1}L_1 = UU_1^{-1} = I(\text{单位阵})$$

即 $L_1 = L$, $U_1 = U$. 唯一性成立.

当 A 分解为 LU 的乘积后,求解方程组 $AX = b$ 就变得很容易了. 因为由 $A = LU$,则方程组 $AX = b$ 就变成 $(LU)X = b$.

如果令 $UX = Y$,则有 $LY = b$

即
$$AX = b \Leftrightarrow \begin{cases} LY = b \\ UX = Y \end{cases}$$

也就是说,求解 $AX = b$ 的问题等价于相继求解二个三角形方程组 $LY = b$ 和 $UX = Y$. 也即

$$\left(\text{下三角}\right) \cdot \begin{pmatrix} \times \\ \vdots \\ \times \end{pmatrix} = \begin{pmatrix} \times \\ \vdots \\ \times \end{pmatrix} \downarrow \text{自上而下}$$

$$\left(\text{上三角}\right) \cdot \begin{pmatrix} \times \\ \vdots \\ \times \end{pmatrix} = \begin{pmatrix} \times \\ \vdots \\ \times \end{pmatrix} \uparrow \text{自下而上}$$

由于三角形方程组很容易求解,因此,高斯消去法的关键在于确定系数矩阵 A 的分解式.

由定理 5.2 还可推得更一般的 A 的 $L\ D\ R$ 分解,即

定理 5.3 设 $A\in R^{n\times n}$，如果 A 的所有顺序主子式均不等于零，则方阵 A 存在唯一的分解式：

$$A=LDR$$

其中，L，R 分别是 n 阶单位下三角阵和单位上三角阵，D 是 n 阶非奇异的对角阵.

事实上，由 $A=LU$，其中

$$U=\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

$$=\begin{bmatrix} u_{11} & & & \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{bmatrix}\begin{bmatrix} 1 & \dfrac{u_{12}}{u_{11}} & \cdots & \dfrac{u_{1n}}{u_{11}} \\ & 1 & \cdots & \dfrac{u_{2n}}{u_{22}} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}=DR$$

且此分解唯一，从而有 $A=LDR$，且分解唯一.

由定理 5.3 知，若在矩阵 A 的唯一分解式 LDR 中改变形式，就可得到 A 的不同分解形式，在计算机上一般有下面三种常用分解方法：

（1）把 D 并入 R：$A=L(DR)=LU$

式中，L 是单位下三角阵，U 是上三角阵. 这种分解即为上面已介绍过的 A 的 LU 直接三角分解，也称为矩阵 A 的杜利脱尔(Doolittle)分解.

（2）把 D 并入 L：$A=(LD)R=\widetilde{L}\widetilde{U}$

式中，$\widetilde{L}$ 是下三角阵，$\widetilde{U}$ 是单位上三角阵. 这种分解称为矩阵 A 的克洛脱(Crout)分解.

（3）若 A 为 n 阶对称正定阵，则由对称正定阵的性质可证明

必有 $A=\widetilde{L}\widetilde{L}^{\mathrm{T}}$ 分解，其中，$\widetilde{L}$ 是对角元均为正的下三角阵. 这种分解称为正定矩阵 A 的乔列斯基(Cholesky)分解.

以上，仅仅从理论上给出 A 的三种矩阵分解，到底具体如何实现分解，分解的递推计算公式如何建立，下面我们将分别进行详细讨论，并给出计算机上常用的算法.

二、直接三角分解法

设方程组 $AX=b$ 的系数矩阵 A 的各阶顺序主子式均不等于零，则由定理 5.2 知，A 可唯一地分解为：$A=LU$. 下面我们利用矩阵的乘法规则，直接从矩阵 A 的元素来推得计算矩阵 L 和 U 的元素的计算公式，从而给出求解线性代数方程组 $AX=b$ 的直接三角分解法.

设 $A=LU$

即
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$=\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & v_{2n} \\ & & & \vdots \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix} \tag{5.11}$$

下面说明 L,U 的元素可以通过 n 步直接由 A 的元素计算定出，其中，第 k 步定出 U 的第 k 行和 L 的第 k 列元素.

由矩阵乘法可知：

(1) 由 L 的第 1 行和 U 的第 j 列元素对应相乘后与 A 的对应元素相等，得

$$a_{1j}=u_{1j}$$

即 U 的第 1 行元素

$$u_{1j}=a_{1j},(j=1,2,\cdots,n)$$

由 L 的第 i 行和 U 的第 1 列元素对应相乘后与 A 的对应元素相等，得

$$a_{i1}=l_{i1}\cdot u_{11}$$

从而得到 L 的第 1 列元素

$$l_{i1}=a_{i1}/u_{11},\ (i=2,3,\cdots,n).$$

这就定出了 U 的第 1 行元素及 L 的第 1 列元素. 类似地，用矩阵乘法再确定 U 的第 2 行及 L 的第 2 列，如此继续……

假设已经定出了 U 的前 $k-1$ 行及 L 的前 $k-1$ 列$(1\leqslant k\leqslant n)$的全部元素，现在来确定 U 的第 k 行元素和 L 的第 k 列元素.

(2) 在式(5.11)中，由矩阵乘法有

$$\underset{(j\geqslant k)}{a_{kj}}=\sum_{r=1}^{n}l_{kr}u_{rj}$$

注意，由于 L 是单位下三角阵，当 $r>k$ 时，$l_{kr}=0$，而 $l_{kk}=1$，所以

$$a_{kj}=\sum_{r=1}^{k-1}l_{kr}\cdot u_{rj}+u_{kj},(j=k,k+1,\cdots,n)$$

从而有

$$u_{kj}=a_{kj}-\sum_{r=1}^{k-1}l_{kr}\cdot u_{rj},(j=k,k+1,\cdots,n)$$

即得到了计算上三角阵 U 的第 k 行元素.

同样道理，可确定 L 的第 k 列元素.

由
$$\underset{(i>k)}{a_{ik}}=\sum_{r=1}^{n}l_{ir}\cdot u_{rk}$$

注意，当 $r>k$ 时，$u_{rk}=0$

则

$$a_{ik}=\sum_{r=1}^{k}l_{ir}\cdot u_{rk}=\sum_{r=1}^{k-1}l_{ir}\cdot u_{rk}+l_{ik}\cdot u_{kk}$$

从而有

$$l_{ik}=(a_{ik}-\sum_{r=1}^{k-1}l_{ir}\cdot u_{rk})/u_{kk},(i=k+1,k+2,\cdots,n)$$

即得到了计算 L 的第 k 列元素.

上述步骤共进行 n 步,就可以定出 L 及 U 的全部元素,完成了矩阵 A 的 LU 分解,也即计算 $A=LU$ 分解的公式为

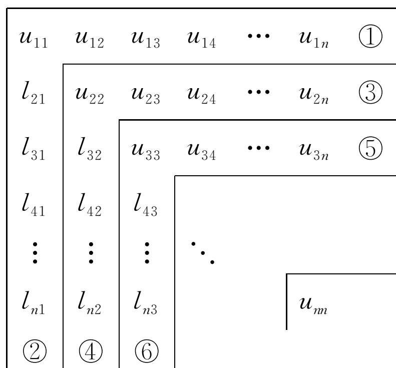
对 $k=1,2,3,\cdots,n$,计算

$$\begin{cases}u_{kj}=a_{kj}-\sum_{r=1}^{k-1}l_{kr}\cdot u_{rj}, & (j=k,k+1,\cdots,n);\\ l_{ik}=(a_{ik}-\sum_{r=1}^{k-1}l_{ir}\cdot u_{rk})/u_{kk},(i=k+1,k+2,\cdots,n).\end{cases}$$

(其中 $\sum_{1}^{0}=0$)

(5.12)

上述这种矩阵 A 的 LU 分解计算顺序也可按下图所示一框一框地逐步进行.



由于以上计算公式(5.12)中不含消去法的中间结果 $a_{ij}^{(k)}$,是可直接逐框计算,所以称为紧凑格式.

现在把矩阵的三角分解方法用来求解方程组 $\boldsymbol{AX}=\boldsymbol{b}$,当系数矩阵 $\boldsymbol{A}$ 完成了 $\boldsymbol{A}=\boldsymbol{LU}$ 分解后,这时,方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 就化为 $\boldsymbol{L}(\boldsymbol{UX})=\boldsymbol{b}$,它等价于求解方程组:

$$\begin{cases}\boldsymbol{LY}=\boldsymbol{b}\\ \boldsymbol{UX}=\boldsymbol{Y}\end{cases}$$

即方程组的求解可分两步完成:

第一步:先解下三角型方程组 $\boldsymbol{LY}=\boldsymbol{b}$

$$\begin{bmatrix}1 & & & & & & \\ l_{21} & 1 & & & & & \\ l_{31} & l_{32} & 1 & & & & \\ \vdots & \vdots & \ddots & \ddots & & & \\ l_{k1} & l_{k2} & & l_{k,k-1} & 1 & & \\ \vdots & \vdots & & \ddots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & & & l_{n,n-1} & 1\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\\ \vdots\\ y_n\end{bmatrix}=\begin{bmatrix}b_1\\ b_2\\ \vdots\\ b_k\\ \vdots\\ b_n\end{bmatrix}\downarrow$$

$\boldsymbol{L}$ 是单位下三角阵,则有

$$\begin{cases}y_1=b_1\\ l_{k1}y_1+l_{k2}y_2+\cdots+l_{k,k-1}y_{k-1}+y_k=b_k\end{cases}\tag{5.13}$$

若令 $\sum_{1}^{0}=0$

则得统一计算公式为

$$y_k=b_k-\sum_{r=1}^{k-1}l_{kr}\cdot y_r,(k=1,2,\cdots,n)\tag{5.14}$$

第二步:再解上三角形方程组 $\boldsymbol{UX}=\boldsymbol{Y}$

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ & \ddots & \vdots & \vdots & \vdots \\ & & u_{kk} & \cdots & u_{kn} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_n \end{bmatrix}$$

从下往上回代,即得递推公式:

$$x_k = (y_k - \sum_{r=k+1}^{n} u_{kr} \cdot x_r)/u_{kk}, (k=n, n-1, \cdots, 2, 1) \quad (5.15)$$

上述这种通过系数矩阵的 LU 分解求解方程组的方法,称为直接三角分解法,也叫做杜利脱尔分解方法.

例 5.2　用直接三角分解法求解方程组

$$\begin{bmatrix} 2 & 2 & 3 \\ 4 & 7 & 7 \\ -2 & 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ -7 \end{bmatrix}$$

解　(1) **分解 $\boldsymbol{A}=\boldsymbol{LU}$**

由公式(5.12)**得**

$$u_{11}=2, u_{12}=2, u_{13}=3,$$

$$l_{21}=a_{21}/u_{11}=4/2=2, l_{31}=-2/2=-1,$$

$$u_{22}=a_{22}-l_{21}u_{12}=3, u_{23}=a_{23}-l_{21}u_{13}=1,$$

$$l_{32}=(a_{32}-l_{31}u_{12})/u_{22}=6/3=2,$$

$$u_{33}=a_{33}-l_{31}u_{13}-l_{32}u_{23}=5+3-2=6$$

即

$$\boldsymbol{A}=\begin{bmatrix} 2 & 2 & 3 \\ 4 & 7 & 7 \\ -2 & 4 & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 2 & 3 \\ 2 & 3 & 1 \\ -1 & 2 & 6 \end{bmatrix}$$

也即
$$A=\begin{bmatrix}1 & & \\ 2 & 1 & \\ -1 & 2 & 1\end{bmatrix}\begin{bmatrix}2 & 2 & 3\\ & 3 & 1\\ & & 6\end{bmatrix}=LU$$

（2）求解 $LY=b$

$$\begin{bmatrix}1 & & \\ 2 & 1 & \\ -1 & 2 & 1\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix}=\begin{bmatrix}3\\ 1\\ -7\end{bmatrix}$$

得到 $Y=(3,-5,6)^{\mathrm{T}}$

（3）求解 $UX=Y$

$$\begin{bmatrix}2 & 2 & 3\\ & 3 & 1\\ & & 6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}3\\ -5\\ 6\end{bmatrix}$$

得到
$$X=(2,-2,1)^{\mathrm{T}}$$

注意，上面所谓的紧凑格式的计算规则不仅用于分解 A，而且能用于在把 $AX=b$ 消元化成 $UX=Y$ 时从 b 算出 Y. 这不妨把 b 看作在 A 右边的另一个列，并按照对待 A 的元 $a_{ij}(i<j)$ 的分解办法一样来处理. 也就是说，按

$$\begin{bmatrix}u_{11} & u_{12} & \cdots & u_{1n} & y_1\\ l_{21} & u_{22} & \cdots & u_{2n} & y_2\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ l_{n1} & l_{n2} & & u_{nn} & y_n\end{bmatrix}$$

计算. 如上例

$$\begin{bmatrix}2 & 2 & 3 & \vdots & 3\\ 4 & 7 & 7 & \vdots & 1\\ -2 & 4 & 5 & \vdots & -7\end{bmatrix}\to\begin{bmatrix}2 & 2 & 3 & \vdots & 3\\ 2 & 3 & 1 & \vdots & -5\\ -1 & 2 & 6 & \vdots & 6\end{bmatrix}$$

由于在计算机计算时，一旦 u_{kj} 被计算出来后，a_{kj} 就失去作用了，为了节约存放单元，可把计算好的 $\boldsymbol{L},\boldsymbol{U}$ 的元素仍然存放在 $\boldsymbol{A}$ 的相应元素的位置上，例如：

$$\boldsymbol{A}=(a_{ij})_{4\times 4}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \rightarrow \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & u_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & u_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & u_{44} \end{bmatrix}$$

最后，在存放 $\boldsymbol{A}$ 的数组中得到 $\boldsymbol{L},\boldsymbol{U}$ 的元素.

用系数矩阵 $\boldsymbol{A}$ 的 $\boldsymbol{LU}$ 直接三角分解法求解方程组所需要的工作量仍为 $\frac{1}{3}n^3+O(n^2)$，和高斯消去法计算同阶方程组所需要的计算工作量基本相同，但是用 $\boldsymbol{LU}$ 分解，把对系数矩阵的计算和对右端项的计算分开了，这就使我们在计算系数矩阵相同而右端项不同的若干方程组

$$\boldsymbol{AX}=(\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_m)$$

时更显得方便（其中 $\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_m$ 是各方程组的右端向量），只须做一次矩阵分解 $\boldsymbol{A}=\boldsymbol{LU}$，然后解 m 个三角形方程组

$$\boldsymbol{UX}=\boldsymbol{Y}_i,\qquad (i=1,2,\cdots,m)$$

且每多解一个方程组仅需增加大约 n^2 次乘（除）法运算.

但必需指出，当 $\boldsymbol{A}$ 为一般非奇异矩阵时，在用直接三角分解法解方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的计算过程中，有可能出现主元素 $u_{kk}=0$ 或 u_{kk} 绝对值很小的情况，使计算严重失真. 因此，为了提高计算解的精度，在三角分解法中也可采用与前面介绍的列主元素消去法类似的选主元的技巧，即得到部分选主元的三角分解法. 详细内容可参看[1].

完全类似地可以推得关于

$$\boldsymbol{A}=\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{U}}$$

的克洛脱分解的递推公式以及用克洛脱分解求解方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 算法（其中 $\widetilde{\boldsymbol{L}}$——下三角阵，$\widetilde{\boldsymbol{U}}$——单位上三角阵），这作为练习，留给大家自己推导.

上面介绍了关于一般 n 阶方阵的三角分解，事实上，在某些特殊情形下，分解的形式和分解的过程还可以简化. 下面再介绍一种关于对称正定矩阵的三角分解.

三、平方根法和改进的平方根法

在科学研究和工程技术的实际计算中遇到的线性代数方程组，其系数矩阵往往具有对称正定的性质. 例如，应用有限元素法解结构力学问题时，最后归结为求解一个线性代数方程组，其系数矩阵大多具有对称正定的性质，对于系数矩阵具有这种特殊性质的方程组，上面介绍的直接三角分解法还可以简化，得到所谓的平方根法，这是计算机上常用的有效方法之一. 下面先讨论对称正定矩阵的三角分解.

设 $\boldsymbol{A}\in \boldsymbol{R}^{n\times n}$ 是 n 阶实矩阵，由代数知识知道，所谓 $\boldsymbol{A}$ 是对称阵，即 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，所谓 $\boldsymbol{A}$ 是正定矩阵，即对于任意 n 维非零列向量 $\boldsymbol{X}\neq 0,\boldsymbol{X}\in \boldsymbol{R}^{n}$，恒有 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{AX}>0$，对称正定阵有许多好的性质. 这里，我们不去一一讨论，只列举一条在下面要用的性质：

若 $\boldsymbol{A}$ 为对称正定阵，则 $\boldsymbol{A}$ 的各阶顺序主子式

$$\Delta_i>0,(i=1,2,\cdots,n)$$

根据这条性质，我们就可以来讨论对称正定矩阵的三角分解，从而给出求解方程组的平方根法.

定理 5.4（对称正定矩阵的三角分解）

如果 $\boldsymbol{A}$ 为对称正定矩阵，则存在一个实的非奇异下三角矩阵 $\widetilde{\boldsymbol{L}}$，使 $\boldsymbol{A}=\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^{\mathrm{T}}$.

且当限定 $\widetilde{\boldsymbol{L}}$ 的对角元素为全正时，这种分解是唯一的.

证明 由 $\boldsymbol{A}$ 的对称正定性，则 $\boldsymbol{A}$ 的各阶顺序主子式 $\Delta_i \neq 0$，$(i=1,2,\cdots,n)$，于是由定理 5.3 可知，$\boldsymbol{A}$ 总存在唯一的 $\boldsymbol{LDR}$ 分解，即 $\boldsymbol{A}=\boldsymbol{LDR}$，其中，$\boldsymbol{L}$ 是单位下三角阵，$\boldsymbol{D}$ 是非奇异的对角阵，$\boldsymbol{R}$ 是单位上三角阵.

又由 $\boldsymbol{A}$ 的对称性，$\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}$，则 $(\boldsymbol{LDR})^{\mathrm{T}}=\boldsymbol{R}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}=\boldsymbol{LDR}$，由分解唯一性，于是有 $\boldsymbol{L}=\boldsymbol{R}^{\mathrm{T}}$，从而得 $\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{T}}$，这表明对称正定阵 $\boldsymbol{A}$ 的 $\boldsymbol{LDR}$ 分解具有特殊形式

$$\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{T}}$$

设 $\boldsymbol{D}=\mathrm{diag}(d_1,d_2,\cdots,d_n)$，$d_j\neq 0$，$(j=1,2,\cdots,n)$.

下面我们进一步证明 $\boldsymbol{D}$ 的对角元均为正数，即 $d_j>0$.

由于 $\boldsymbol{L}$ 是单位下三角矩阵，所以对于单位坐标向量，

$$\boldsymbol{e}_j=(0,\cdots,0,\underset{(j)}{1},0,\cdots,0)^{\mathrm{T}}$$

存在非零向量 $\boldsymbol{X}_j$，使

$$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{X}_j=\boldsymbol{e}_j,(j=1,2,\cdots,n)$$

因此，$\boldsymbol{X}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}_j=\boldsymbol{X}_j^{\mathrm{T}}(\boldsymbol{LDL}^{\mathrm{T}})\boldsymbol{X}_j=(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{X}_j)^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{X}_j)=\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{D}\boldsymbol{e}_j=d_j$

根据 $\boldsymbol{A}$ 是对称正定阵的定义，有 $\boldsymbol{X}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}_j>0$，

从而，$d_j>0$，$(j=1,2,\cdots,n)$

这就证明了 $\boldsymbol{D}$ 的对角元皆为正数.

现设 $$\boldsymbol{D}^{1/2}=\mathrm{diag}[\sqrt{d_1},\sqrt{d_2},\cdots,\sqrt{d_n}]$$

注意，在这里我们将 $\boldsymbol{D}^{1/2}$ 的对角元全取为正数，即

$$\boldsymbol{D}=\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix}$$

$$
=\begin{bmatrix}\sqrt{d_1} & & & \\ & \sqrt{d_2} & & \\ & & \ddots & \\ & & & \sqrt{d_n}\end{bmatrix}\begin{bmatrix}\sqrt{d_1} & & & \\ & \sqrt{d_2} & & \\ & & \ddots & \\ & & & \sqrt{d_n}\end{bmatrix}
$$

则
$$
\begin{aligned}
\boldsymbol{A} &= \boldsymbol{LDL}^{\mathrm{T}} = \boldsymbol{LD}^{1/2}\boldsymbol{D}^{1/2}\boldsymbol{L}^{\mathrm{T}} \\
&= (\boldsymbol{LD}^{1/2})(\boldsymbol{LD}^{1/2})^{\mathrm{T}} = \widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^{\mathrm{T}}
\end{aligned}
$$

其中,$\widetilde{\boldsymbol{L}}=\boldsymbol{LD}^{1/2}$,显然是对角元全为正的非奇异的下三角阵.

由于分解式 $\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{T}}$ 是唯一的. 又限定 $\boldsymbol{D}^{1/2}$ 的对角元为正数,从而分解 $\boldsymbol{D}=\boldsymbol{D}^{1/2}\boldsymbol{D}^{1/2}$ 也是唯一的,所以说,在限定 $\boldsymbol{L}$ 的对角元皆为正数时,三角分解是唯一的.

对称正定阵 $\boldsymbol{A}$ 的三角分解 $\boldsymbol{A}=\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^{\mathrm{T}}$ 称为正定矩阵 $\boldsymbol{A}$ 的乔列斯基(Cholesky)分解,又称为 LLT 分解.

下面我们先用直接分解法来确定计算 $\widetilde{\boldsymbol{L}}$ 的元素的递推公式,然后,再给出用平方根法解线性代数方程组的计算公式.

设矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$ 是 n 阶对称正定矩阵,它的三角分解为 $\boldsymbol{A}=\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^{\mathrm{T}}$,为了方便,常记为

$$
\boldsymbol{A}=\boldsymbol{LL}^{\mathrm{T}} \tag{5.16}
$$

这里

$$
\boldsymbol{L}=\begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn}\end{bmatrix}
$$

为下三角阵,且其中对角元 $l_{ii}>0$,$(i=1,2,\cdots,n)$.

利用矩阵的乘法规则,可直接由 $\boldsymbol{A}$ 的元素 a_{ij} 逐行或逐列确定 $\boldsymbol{L}$ 的元素. 下面我们介绍按列自左向右依次确定 $\boldsymbol{L}$ 的元素的计算

公式.

（1）计算 L 的第 1 列元素 l_{i1},$(i=1,2,\cdots,n)$

由矩阵乘法知,$a_{11}=l_{11}\cdot l_{11}$,$a_{i1}=l_{i1}\cdot l_{11}$,$(i>1)$

即得 $l_{11}=\sqrt{a_{11}}$,(由于 $a_{11}>0$,开方有意义)

$$l_{i1}=a_{i1}/l_{11},(i=2,3,\cdots,n) \tag{5.17}$$

（2）计算 L 的第 2 列元素 l_{i2},$(i=2,3,\cdots,n)$

由矩阵乘法知

$$a_{22}=l_{21}^2+l_{22}^2$$

所以 $$l_{22}=\sqrt{a_{22}-l_{21}^2}$$

又 $$a_{i2}=l_{i1}l_{21}+l_{i2}l_{22},\quad (i>2)$$

所以 $$l_{i2}=(a_{i2}-l_{i1}l_{21})/l_{22} \tag{5.18}$$

这样就定出了 L 的第一列元素和第二列元素,如此继续下去,假定已算出了 L 的前 $k-1$ 列$(1\leqslant k\leqslant n)$,现在来确定 L 的第 k 列元素.

（3）计算 L 的第 k 列元素 $l_{ik}$$(i=k,k+1,\cdots,n)$

由矩阵乘法知:

$$a_{kk}=l_{k1}^2+l_{k2}^2+\cdots+l_{kk}^2$$

所以 $$l_{kk}=\sqrt{a_{kk}-\sum_{r=1}^{k-1}l_{kr}^2}$$

又 $$\underset{(i>k)}{a_{ik}}=l_{i1}\cdot l_{k1}+l_{i2}\cdot l_{k2}+\cdots+l_{i,k-1}l_{k,k-1}+l_{ik}l_{kk}$$

所以 $$l_{ik}=(a_{ik}-\sum_{r=1}^{k-1}l_{ir}l_{kr})/l_{kk}$$

$$(i=k+1,k+2,\cdots,n) \tag{5.19}$$

这样,就定出了 L 的第 k 列元素.

上述步骤进行 n 步，就可以定出 $\boldsymbol{L}$ 的全部元素，从而实现了 $\boldsymbol{A}$ 的 $\boldsymbol{LL}^{\mathrm{T}}$ 分解，其计算工作量只是一般直接三角分解的一半多一点，且由

$$a_{kk}=\sum_{r=1}^{k} l_{kr}^{2}$$

可知 $$|l_{kr}|\leqslant a_{kk}^{1/2},(r=1,2,\cdots,n;k=1,2,\cdots,n)$$

这表明 $\boldsymbol{L}$ 的元素的绝对值一般不会很大，所以计算是稳定的，这是乔列斯基分解的又一个优点.

利用系数矩阵的乔列斯基分解，求解对称正定方程组的方法，称为平方根法. 若实现了对称正定矩阵的乔列斯基分解 $\boldsymbol{A}=\boldsymbol{LL}^{\mathrm{T}}$，则解方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 即求解方程组 $(\boldsymbol{LL}^{\mathrm{T}})\boldsymbol{X}=\boldsymbol{b}$，等价于求解两个三角形方程组：$\boldsymbol{LY}=\boldsymbol{b}$ 和 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{X}=\boldsymbol{Y}$.

综上所述，用平方根法求解对称正定方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的计算步骤可归纳为：

(1) 对矩阵 $\boldsymbol{A}$ 进行乔列斯基分解：$\boldsymbol{A}=\boldsymbol{LL}^{\mathrm{T}}$，按列确定下三角阵 $\boldsymbol{L}$，即对于 $k=1,2,\cdots,n$，计算

① $$l_{kk}=(a_{kk}-\sum_{r=1}^{k-1} l_{kr}^{2})^{1/2}$$

② $$l_{ik}=(a_{ik}-\sum_{r=1}^{k-1} l_{ir}l_{kr})/l_{kk} \tag{5.20}$$

$$(i=k+1,k+2,\cdots,n)$$

(2) 求解下三角形方程组 $\boldsymbol{LY}=\boldsymbol{b}$. 即

$$y_{k}=(b_{k}-\sum_{r=1}^{k-1} l_{kr}y_{r})/l_{kk} \tag{5.21}$$

$$(k=1,2,\cdots,n),(\text{其中}\sum_{r=1}^{0}=0)$$

(3) 求解上三角形方程组 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{X}=\boldsymbol{Y}$. 即

$$x_k=(y_k-\sum_{r=k+1}^{n} l_{rk}x_r)/l_{kk} \tag{5.22}$$

$$(k=n,n-1,\cdots,2,1),(\text{其中}\sum_{r=n+1}^{n}=0)$$

用平方根法求解对称正定方程组是目前在计算机上广泛采用的行之有效的方法.

例 5.3 求解方程组

$$\begin{bmatrix} 4 & 2 & -2 \\ 2 & 2 & -3 \\ -2 & -3 & 14 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 10 \\ 5 \\ 4 \end{bmatrix}$$

解 系数矩阵 $\boldsymbol{A}$ 显然是对称正定的,由 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 的分解公式(5.20)得

$$l_{11}=\sqrt{4}=2,\ l_{21}=2/2=1,\ l_{31}=-2/2=-1,$$

$$l_{22}=\sqrt{2-1}=1,\ l_{32}=-2,\ l_{33}=\sqrt{14-1^2-2^2}=3$$

因此

$$\boldsymbol{L}=\begin{bmatrix} 2 & & \\ 1 & 1 & \\ -1 & -2 & 3 \end{bmatrix}$$

于是由公式(5.21)求解 $\boldsymbol{L}\boldsymbol{Y}=\boldsymbol{b}$

解得 $\boldsymbol{Y}=(5,0,3)^{\mathrm{T}}$

再由公式(5.22)求解 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{X}=\boldsymbol{Y}$ 得方程组的解

$$\boldsymbol{X}=(2,2,1)^{\mathrm{T}}$$

用平方根法解对称正定方程组,在计算 $\boldsymbol{L}$ 的对角元素 $l_{kk}(k=1,2,\cdots,n)$时需要进行 n 次开方运算,为了避免开方运算,我们可直接采用对称正定矩阵的 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 分解式对平方根法进行改

进，这里 L 为单位下三角阵，D 为非奇异对角阵.

从而解方程组 $AX=b$ 可按如下步骤进行：

(1) 把系数矩阵 A 分解成

$$A=LDL^{\mathrm{T}}$$

则 $AX=b$ 变成 $(LDL^{\mathrm{T}})X=b$，此方程组等价于相继求解两个方程组：

$$\begin{cases} LY=b \\ L^{\mathrm{T}}X=D^{-1}Y \end{cases}$$

(2) 解下三角形方程组 $LY=b$，求出 Y.

(3) 解上三角形方程组 $L^{\mathrm{T}}X=D^{-1}Y$，求出 X.

上述方法称为改进的平方根法，它比较平方根法的优点是回避了开平方运算.

解对称正定方程组的平方根法或改进的平方根法，是目前在计算机上解对称正定方程组的一个有效方法，应用较广泛，其计算量和存贮量都比用消去法约节省近一半，且不需要选主元，便能求得较精确的数值解.

5.3 解带状方程组的三角分解法

在很多实际问题中，例如，在结构分析，网络理论和微分方程数值解等问题中，都会遇到求解大型带状线性方程组，其系数矩阵的特点是含有大量的零元素，且其非零元素均分布在对角线元素的两侧成带状，对于这类特殊线性代数方程组，也可采用三角分解法，且能经济有效地实现其求解运算.

定义 5.1 设 $A=(a_{ij})_{n\times n}\in \boldsymbol{R}^{n\times n}$，对小于 n 的正整数 p 和 q，当 $j>i+q$ 或 $i>j+p$ 时，$a_{ij}=0$，则称 A 是具有上带宽 q 和下带宽 p 的带状矩阵，以带状矩阵为系数矩阵的方程组 $AX=b$，称为

带状方程组.特别,当 $p=q=t$ **时,称** t **为** $\boldsymbol{A}$ **的半带宽,** $2t-1$ **称为** $\boldsymbol{A}$ **的总带宽.**

一般, n **较大,且** $p,q\ll n$**.带状矩阵的** $\boldsymbol{LU}$ **分解只需计算非零元素,且可证明** $\boldsymbol{L}$ **和** $\boldsymbol{U}$ **仍保持** $\boldsymbol{A}$ **的带状结构,即有**

定理 5.5(保带状结构定理)

设(1) $\boldsymbol{A}=(a_{ij})_{n\times n}\in\boldsymbol{R}^{n\times n}$ **是上、下带宽分别为** q **和** p **的带状矩阵;**

(2) $\boldsymbol{A}$ **的所有顺序主子式** $\Delta_k\neq 0,(k=1,2,\cdots,n)$;

则 $\boldsymbol{A}$ **存在唯一的三角分解式:**

$$\boldsymbol{A}=\boldsymbol{LU}$$

其中, $\boldsymbol{L}$ **是下半带宽为** p **的单位下三角阵,** $\boldsymbol{U}$ **是上半带宽为** q **的上三角阵.**

证明　设 $a_{ij}=0$ **为** $\boldsymbol{A}$ **的带外元素**($i>j$ **且** $i>j+p$)**,于是有**

$$a_{i1}=a_{i2}=\cdots=a_{i,j-1}=a_{ij}=0$$

由 5.2 **节** $\boldsymbol{A}=\boldsymbol{LU}$ **的分解公式**(5.12)**得单位下三角阵** $\boldsymbol{L}$ **的元素**

$$l_{ij}=(a_{ij}-\sum_{r=1}^{j-1}l_{ir}\cdot u_{rj})/u_{jj} \tag{5.23}$$

则有　$l_{i1}=a_{i1}/u_{11}=0$,　$l_{i2}=(a_{i2}-l_{i1}u_{12})/u_{22}=0,\cdots,$

$l_{i,j-1}=0, l_{ij}=0$.

同理可证,当 $a_{ij}=0,(j>i$ **且** $j>i+q)$**时,有** $u_{ij}=0$**,即** $\boldsymbol{L}$ **和** $\boldsymbol{U}$ **分别是带宽为** p **和** q **的下、上三角矩阵.**

根据定理 5.5,A 的 LU 分解可以像 5.2 节公式(5.12)一样计算,只是不需要计算 L 和 U 中的零元,即对 L 中的第 k 列,只需计算第 $k+1$ 至 $k+p$ 行的元;对 U 的第 k 行只需计算第 $k+1$ 到 $k+q$ 列的元,从而,当 p 和 q 都不大时,其乘除法的计算工作量比 $n^3/3$小得多.

一、用三角分解法解大型等带宽方程组

设 $AX=b$,其中,$A=(a_{ij})_{n\times n}\in R^{n\times n}$ 是半带宽为 t 的带状矩阵,且 A 的顺序主子式$\Delta_k\neq 0,(k=1,2,\cdots,n)$. 则由定理 5.5,$A$ 存在唯一的三角分解式:$A=LU$,且 L 和 U 保持 A 的带状结构,其带宽均为 t. 则求解方程组 $AX=b$ 的计算公式完全类同于 5.2 节介绍的杜利脱尔分解法中的公式(5.12)、公式(5.14)和公式(5.15),只是公式中对 L,U 的带外零元素不必计算,且 L,U 带外零元素可以不必参加求和运算,稍作修改,即可得如下用三角分解法解大型等带宽方程组的计算公式:

(1) 分解 $A=LU$

对 $k=1,2,\cdots,n$,计算

$$\begin{cases} u_{kj}=a_{kj}-\sum\limits_{r=\max(1,j-t)}^{k-1} l_{kr}\cdot u_{rj}, & (j=k,k+1,\cdots,\min(k+t,n)) \\ l_{ik}=(a_{ik}-\sum\limits_{r=\max(1,i-t)}^{k-1} l_{ir}\cdot u_{rk})/u_{kk}, & (i=k+1,k+2,\cdots,\min(k+t,n)) \end{cases}$$

(其中 $\sum\limits_{1}^{0}=0$) (5.24)

(2) 求解

① 解 $LY=b$

$$\begin{cases} y_1=b_1 \\ y_k=b_k-\sum\limits_{r=\max(1,k-t)}^{k-1} t_{kr}\cdot y_r,(k=2,3,\cdots,n) \end{cases} \tag{5.25}$$

② 解 $\boldsymbol{UX}=\boldsymbol{Y}$

$$\begin{cases} x_n = y_n / u_{nn} \\ x_k = (y_k - \sum\limits_{r=k+1}^{\min(k+t,n)} u_{kr} \cdot x_r) / u_{kk}, (k=n-1,\cdots,2,1) \end{cases} \tag{5.26}$$

二、解三对角方程组的追赶法

当带宽 $p=q=1$ 时，等带宽矩阵称为三对角矩阵，三对角矩阵是最简单，然而又是经常遇到的带状矩阵，例如，在建立三次样条函数，求微分方程数值解等问题中，都会遇到求解这样的线性代数方程组：

$$\boldsymbol{AX}=\boldsymbol{d} \tag{5.27}$$

它的系数方阵 $\boldsymbol{A}$ 是一个阶数较高的三对角线方阵

$$\boldsymbol{A}=\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix},$$

简记为 $\boldsymbol{A}=[a_i,b_i,c_i]_1^n$，右端项 $\boldsymbol{d}=(d_1,d_2,\cdots,d_n)^{\mathrm{T}}$. 称式(5.27)为三对角方程组.

现在我们也从矩阵的三角分解来讨论三对角线性方程组(5.27)的求解.

假定系数矩阵 $\boldsymbol{A}=[a_i,b_i,c_i]_1^n$ 存在克洛脱(Crout)分解：

$$\boldsymbol{A}=\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{U}} \tag{5.28}$$

由于 $\boldsymbol{A}$ 是带宽为 1 的三对角阵，所以，$\widetilde{L}$，$\widetilde{U}$ 也具有特殊的形式，均为二对角阵，即

$$\widetilde{\boldsymbol{L}}=\begin{bmatrix} l_1 & & & & \\ m_2 & l_2 & & & \\ & \ddots & \ddots & & \\ & & m_{n-1} & l_{n-1} & \\ & & & m_n & l_n \end{bmatrix}$$

$$\widetilde{\boldsymbol{U}}=\begin{bmatrix} 1 & u_1 & & & \\ & 1 & u_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & u_{n-1} \\ & & & & 1 \end{bmatrix}$$

在等式(5.28)**右端利用矩阵乘法,然后与左端比较,便可得到如下确定 $\widetilde{L}$ 与 $\widetilde{U}$ 的元素** m_i, l_i, u_i **的计算公式**:

$$\begin{cases} l_1=b_1, u_1=\dfrac{c_1}{l_1} \\ m_i=a_i, (i=2,3,\cdots,n) \\ l_i=b_i-m_iu_{i-1}, (i=2,3,\cdots,n) \\ u_i=c_i/l_i, (i=2,3,\cdots,n-1) \end{cases} \tag{5.29}$$

对三对角线性方程组的系数矩阵 $\boldsymbol{A}$ 有了分解式(5.28)**以后,求解 $\boldsymbol{AX}=\boldsymbol{d}$ 的问题就转化为求解两个特殊的三角形方程组**:

(1) $\widetilde{\boldsymbol{L}}\boldsymbol{Y}=\boldsymbol{d}$

(2) $\widetilde{\boldsymbol{U}}\boldsymbol{X}=\boldsymbol{Y}$

由于这两个方程组都是三角形方程组,所以很容易得到其递推的计算式分别为

$$\begin{cases} y_1=d_1/l_1 \\ y_i=(d_i-m_iy_{i-1})/l_i, (i=2,3,\cdots,n) \end{cases} \tag{5.30}$$

及

$$\begin{cases} x_n = y_n \\ x_i = y_i - u_i x_{i+1}, (i=n-1, n-2, \cdots, 1) \end{cases} \tag{5.31}$$

人们常把足标从小到大求解 $y_1 \to y_2 \to \cdots \to y_n$ 的过程式(5.30)形象化地称为“追”的过程;将足标自大到小计算方程组的解 $x_n \to x_{n-1} \to \cdots \to x_2 \to x_1$ 的过程式(5.31)称为“赶”的过程. 因此,上述所给出的解法通常称为追赶法.

从上面的讨论我们看到追赶法的实质就是克洛脱分解法用到求解三对角线性方程组上去的结果. 这时,由于 $\boldsymbol{A}$ 的特殊性,求解运算中将系数矩阵中大量零元素撇开,仅在三条对角线上进行,从而使求解的计算公式特别简单.

综上所述,求解三对角线性方程组的追赶法的计算步骤归结为:

(1) 由公式(5.29)确定三角分解 $\boldsymbol{A}=\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$ 中 $\tilde{\boldsymbol{L}}$ 和 $\tilde{\boldsymbol{U}}$ 的元素;

(2) 由公式(5.30)求出中间向量 $\boldsymbol{Y}$;

(3) 由公式(5.31)求出解向量 $\boldsymbol{X}$.

三对角线性方程组的系数矩阵 $\boldsymbol{A}=[a_i, b_i, c_i]_1^n$ 应满足什么条件才存在形如式(5.28)的克洛脱分解?下面给出一个充分条件:

在三对角矩阵 $\boldsymbol{A}=[a_i, b_i, c_i]_1^n$ 中,若

① $a_i \neq 0$,$(i=2,3,\cdots,n)$ 且 $c_i \neq 0, (1,2,\cdots,n-1)$;

② $|b_1| > |c_1| > 0$

$|b_i| \geqslant |a_i| + |c_i|, (i=2,3,\cdots,n-1)$

$|b_n| > |a_n| > 0$;

则 $\boldsymbol{A}$ 非奇异且存在唯一的形如式(5.28)的分解.

从上面的计算公式得知,用追赶法求解三对角方程组的计算量小,仅需 $5n-4$ 次乘除法运算,且计算是稳定的,存贮量也小.

例 5.4 用追赶法求解三对角线性代数方程组

$$\begin{bmatrix} 2 & -1 & & \\ -1 & 3 & -2 & \\ & -2 & 4 & -3 \\ & & -3 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 6 \\ 1 \\ -2 \\ 1 \end{bmatrix}$$

解　由公式(5.29)**得分解式** $A=\widetilde{L}\widetilde{U}$ **的元素为**

$$m_2=a_2=-1,\quad l_1=b_1=2,\quad u_1=c_1/l_1=-1/2;$$

$$m_3=a_3=-2,\quad l_2=b_2-m_2u_1=5/2,\quad u_2=c_2/l_2=-4/5;$$

$$m_4=a_4=-3,\quad l_3=b_3-m_3u_2=12/5,\quad u_3=c_3/l_3=-5/4;$$

$$l_4=b_4-m_4u_3=5/4.$$

追的过程为

$$y_1=d_1/l_1=3,\ y_2=(d_2-m_2y_1)/l_2=8/5,$$

$$y_3=(d_3-m_3y_2)/l_3=1/2,\ y_4=(d_4-m_4y_3)/l_4=2.$$

赶的过程为

$$x_4=y_4=2,\ x_3=y_3-u_3x_4=3,x_2=y_2-u_2x_3=4,$$

$x_1=y_1-u_1x_2=5.$

所以方程组的解为

$$\boldsymbol{X}=(5,4,3,2)^{\mathrm{T}}.$$

5.4　范数与方程组的状态

为了研究线性代数方程组近似解的误差分析和迭代法的收敛性,需要对向量和矩阵的“大小”引进某种度量——向量和矩阵的范数概念,它们在数值计算中起着重要作用.

一、向量的范数

定义 5.2 设向量 $\boldsymbol{X}\in\boldsymbol{R}^n$，若 $\boldsymbol{X}$ 的某个实值函数 $N(\boldsymbol{X})=\|\boldsymbol{X}\|$ 满足条件：

(1) 非负性：$\|\boldsymbol{X}\|\geqslant 0$ 且 $\|\boldsymbol{X}\|=0$ 的充分必要条件是 $\boldsymbol{X}=0$；

(2) 齐次性：$\|k\boldsymbol{X}\|=|k|\,\|\boldsymbol{X}\|$（$k$ 为任意实数）；

(3) 三角不等式：对任意 $\boldsymbol{X},\boldsymbol{Y}\in\boldsymbol{R}^n$，都有

$$\|\boldsymbol{X}+\boldsymbol{Y}\|\leqslant\|\boldsymbol{X}\|+\|\boldsymbol{Y}\|;$$

则称 $N(\boldsymbol{X})=\|\boldsymbol{X}\|$ 为 $\boldsymbol{R}^n$ 上的向量 $\boldsymbol{X}$ 的范数.

可见，向量的范数其实不过是以实的 n 维向量空间 $\boldsymbol{R}^n$ 中的一切 n 维向量为定义域的一种特殊函数——满足条件(1)，(2)，(3)的一个实的非负函数.

一个向量空间可以定义多种范数，下面我们给出向量 $\boldsymbol{X}$ 的三种常用范数定义.

定义 5.3 设 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\boldsymbol{R}^n$，则定义

(1) 向量的“2-范数”

$$\|\boldsymbol{X}\|_2=\sqrt{x_1^2+x_2^2+\cdots+x_n^2}$$

(2) 向量的“∞-范数”

$$\|\boldsymbol{X}\|_\infty=\max_{1\leqslant i\leqslant n}\{|x_i|\}$$

(3) 向量的“1-范数”

$$\|\boldsymbol{X}\|_1=\sum_{i=1}^{n}|x_i|$$

容易验证，这样定义的向量 $\boldsymbol{X}$ 的三种范数确实满足向量范数的三个条件，因此，它们都是 $\boldsymbol{R}^n$ 上的向量范数，且它们可以统一地记为

$$\|\boldsymbol{X}\|_p=\left[\sum_{i=1}^{n}|x_i|^p\right]^{1/p},\qquad (p=1,2,\infty)$$

并称为向量 X 的"p-范数".

为了方便,有时在不针对这个或那个向量时,我们常把范数记号中的变量省掉,简记为 $\|\cdot\|_p$,甚至写成 $\|\quad\|_p$,$(p=1,2,\infty)$.

例 5.5 计算向量 $X=(1,0,-1,2)^{\mathrm{T}}$ 的三种常用范数.

解 $\|X\|_1=|1|+|0|+|-1|+|2|=4$

$$\|X\|_2=\sqrt{1^2+0^2+(-1)^2+(2)^2}=\sqrt{6}$$

$$\|X\|_\infty=\max\{1,0,|-1|,2\}=2$$

定理 5.6(范数的等价性) 对任何向量 $X,Y\in R^n$,有

(1) $\|X\|_2\leqslant\|X\|_1\leqslant\sqrt{n}\|X\|_2$;

(2) $\|X\|_\infty\leqslant\|X\|_1\leqslant n\|X\|_\infty$;

(3) $\|X\|_\infty\leqslant\|X\|_2\leqslant\sqrt{n}\|X\|_\infty$.

下面证明不等式(2):

证明 由 $\|X\|_\infty=\max\limits_{1\leqslant i\leqslant n}|x_i|$, $\|X\|_1=\sum\limits_{i=1}^{n}|x_i|$

得
$$\max_{1\leqslant i\leqslant n}|X_i|\leqslant|x_1|+|x_2|+\cdots+|x_n|\leqslant n\max_i|x_i|$$

即
$$\|X\|_\infty\leqslant\|X\|_1\leqslant n\|X\|_\infty.$$

有了向量范数的概念,可进一步定义向量序列的收敛性.

定义 5.4 在 R^n 中的一个向量序列 $X^{(1)},X^{(2)},\cdots,X^{(k)}\cdots$ 称为收敛于一个向量 X,当且仅当 $\lim\limits_{k\to\infty}\|X-X^{(k)}\|=0$.

二、矩阵的范数

定义 5.5 设矩阵 $A\in R^{n\times n}$,若 A 的某个实值函数 $N(A)=\|A\|$ 满足条件:

(1) **非负性**:$\|A\|\geqslant0$ 且 $\|A\|=0$ 的充分必要条件是 $A=0$;

(2) **齐次性**:$\|kA\|=|k|\|A\|$(k 为任意实数);

(3) **三角不等式:对任意 $\boldsymbol{A},\boldsymbol{B}\in \boldsymbol{R}^{n\times n}$,都有**
$\|\boldsymbol{A}+\boldsymbol{B}\| \leqslant \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$;

(4) $\|\boldsymbol{AB}\| \leqslant \|\boldsymbol{A}\| \ \|\boldsymbol{B}\|$;

则称 $N(\boldsymbol{A})=\|\boldsymbol{A}\|$ **为** $\boldsymbol{R}^{n\times n}$**上的矩阵** A **的范数.**

在应用问题中,矩阵和向量的乘积经常出现,因而应让所用的矩阵范数与向量范数有某种关系.

定义 5.6　对于给定的向量范数 $\|\cdot\|$ **和矩阵范数** $\|\cdot\|$,**如果对任一个向量** $\boldsymbol{X}\in \boldsymbol{R}^{n}$ **和任一个矩阵** $A\in \boldsymbol{R}^{n\times n}$**都有不等式**

$$\|\boldsymbol{AX}\| \leqslant \|\boldsymbol{A}\| \ \|\boldsymbol{X}\| \text{ 成立},$$

则称所给的矩阵范数与向量范数是相容的.

下面给出一种定义矩阵范数的方法,它是由向量范数诱导出来的,且这种矩阵范数和向量范数是相容的.

定义 5.7　设向量 $X\in \boldsymbol{R}^{n}$,**矩阵** $A\in \boldsymbol{R}^{n\times n}$,**且给定一种向量范数** $\|\boldsymbol{X}\|_p$,**则定义**

$$\|\boldsymbol{A}\|_p=\max_{X\neq 0}\frac{\|\boldsymbol{AX}\|_p}{\|\boldsymbol{X}\|_p},(p=1,2,\infty) \tag{5.32}$$

为矩阵 A **的范数,并称为** A **的算子范数.**

显然,由公式(5.32)**定义的** A **的范数** $\|\boldsymbol{A}\|_p$ **满足方阵范数定义中的所有条件,称其为** $\boldsymbol{R}^{n\times n}$ **上从属于向量范数** $\|\cdot\|_p$ **的矩阵范数,或称为由向量范数** $\|\cdot\|_p$ **导出的矩阵范数,也称为矩阵** A **的"**p**-范数".**

下面分别给出从属于向量1-**范数**,2-**范数及**∞-**范数的三种常用的方阵范数.**

定理 5.7　设 A **是** n **阶方阵:**$\boldsymbol{A}=(a_{ij})_{n\times n}$,
则

$$(1)\ \|\boldsymbol{A}\|_1=\max_{1\leqslant j\leqslant n}\sum_{i=1}^{n}|a_{ij}|; \tag{5.33}$$

(2) $\|\boldsymbol{A}\|_2=$(方阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 之最大特征值)$^{1/2}$

$$=\sqrt{\lambda\max(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})};\tag{5.34}$$

(3) $$\|\boldsymbol{A}\|_\infty=\max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|a_{ij}|.\tag{5.35}$$

$\|\boldsymbol{A}\|_1$ 是 A 的各列元素绝对值之和的最大值,所以 $\|\boldsymbol{A}\|_1$ 又叫做 A 的列范数.

$\|\boldsymbol{A}\|_\infty$ 是 A 的各行元素绝对值之和的最大值,所以 $\|\boldsymbol{A}\|_\infty$ 又叫做 A 的行范数.

定理中的三个公式均可以得到证明.

显然,对于上述三种从属范数,均有 $\|\boldsymbol{I}\|_p=1$,($p=1,2,\infty$),此处 $\boldsymbol{I}$ 为 n 阶单位方阵.

例 5.6 计算 $\boldsymbol{A}=\begin{pmatrix}1 & -2\\ -3 & 4\end{pmatrix}$ 的三种范数 $\|\boldsymbol{A}\|_p$,($p=1,2,\infty$).

解 按公式(5.33),(5.34),(5.35)计算得

$$\|\boldsymbol{A}\|_1=\max\{4,6\}=6$$

$$\|\boldsymbol{A}\|_2=\sqrt{15+\sqrt{221}}\approx 5.46$$

$$\|\boldsymbol{A}\|_\infty=\max\{3,7\}=7.$$

三、谱半径、谱范数与方阵的 F-范数

定义 5.8 设 n 阶方阵 A 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$,则称

$$\rho(\boldsymbol{A})=\max_{1\leqslant i\leqslant n}|\lambda_i|$$

为 A 的谱半径($|\lambda_i|$ 是 λ_i 的模).

不难证明,矩阵的谱半径与范数有着密切关系.

定理 5.8 对于 $\|\cdot\|_p$ ($p=1,2,\infty$)都有

$$\rho(\boldsymbol{A})\leqslant\|\boldsymbol{A}\|_p$$

证明 对于 n 阶方阵 A，设 $\lambda_1=\max\limits_{1\leqslant i\leqslant n}|\lambda_i|$，即 $\rho(\boldsymbol{A})=\lambda_1$，对应特征向量为 $\boldsymbol{u}_1$，则

$$\lambda_1\boldsymbol{u}_1=\boldsymbol{A}\boldsymbol{u}_1\text{，且 }\boldsymbol{u}_1\text{ 为非零向量.}$$

由范数的定义及矩阵范数与向量范数的相容性，有

$$|\lambda_1|\ \|\boldsymbol{u}_1\|=\|\lambda_1\boldsymbol{u}_1\|=\|\boldsymbol{A}\boldsymbol{u}_1\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{u}_1\|,$$

而 $\|\boldsymbol{u}_1\|>0$，不等式两边消去 $\|\boldsymbol{u}_1\|$

即得
$$|\lambda_1|\leqslant\|\boldsymbol{A}\|$$

于是有
$$\rho(\boldsymbol{A})=|\lambda_1|\leqslant\|\boldsymbol{A}\|$$

对 $\|\cdot\|$ 来说，不等式

$$\|\boldsymbol{A}\|_2\geqslant\rho(\boldsymbol{A})$$

在 A 为实对称的情况下成为等式，即

$$\|\boldsymbol{A}\|_2=\rho(\boldsymbol{A})$$

这是因为
$$\|\boldsymbol{A}\|_2^2=\rho(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})=\rho(\boldsymbol{A}^2)=[\rho(\boldsymbol{A})]^2$$

由于 $\|\boldsymbol{A}\|_2=\sqrt{\rho(\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{A})}$，所以 $\|\boldsymbol{A}\|_2$ 又称为谱范数. 且它是与向量的 2-范数相容的方阵范数，所以它又特别重要，但是 $\|\boldsymbol{A}\|_2$ 却不像 $\|\boldsymbol{A}\|_\infty$ 与 $\|\boldsymbol{A}\|_1$ 那样容易计算. 于是，人们希望找到既能与向量的 2-范数相容又容易计算的方阵范数，从而引进 $\boldsymbol{F}$-范数的概念.

定义 5.9 设矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}\in\boldsymbol{R}^{n\times n}$

则称
$$\|\boldsymbol{A}\|_F=\left[\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^2\right]^{1/2}$$

为 A 的 Frobenius 范数，简称为 $\boldsymbol{F}$-范数.

可以证明，$\|\boldsymbol{A}\|_F$ 除了满足方阵范数定义的四个条件外，还

具备

$$\|AX\|_2 \leqslant \|A\|_F \cdot \|X\|_2$$

所以 $\|A\|_F$ 是与向量的 2-范数相容的，但它不是由向量的 2-范数导出的. 要注意，对于 n 阶单位方阵 I 来说

$$\|I\|_F = \sqrt{n}.$$

四、方程组的状态与条件数

用数值计算方法解线性代数方程组的计算结果，有时出现不准确，原因可能有两种：一种情况是计算方法不合理；另一种情况可能是线性代数方程组本身的问题. 对于有问题的方程组，即使用最好的数值方法去求解也是毫无意义的. 下面我们将进一步讨论方程组本身的"好"和"坏"的问题，也即方程组的状态问题.

我们已经知道，一个线性代数方程组 $AX=b$ 完全由它的系数矩阵 A 和右端向量 b 所确定，当 A，b 确定后，X 也就完全确定. 但是，在求解线性代数方程组时，A，b 的数据都是由实际问题所提供的，因而或多或少总有误差，即有扰动，从而使计算结果产生误差，因此需要研究方程组的 A 和 b 的扰动对解 X 的影响.

例 5.7　对下面两个线性代数方程组

$$\begin{cases} 2x_1 + 6x_2 = 8 \\ 2x_1 + 6.00001x_2 = 8.00001 \end{cases}$$

和

$$\begin{cases} 2x_1 + 6x_2 = 8 \\ 2x_1 + 5.99999x_2 = 8.00002 \end{cases}$$

容易算出第一个方程组的解为 $X=(1,1)^{\mathrm{T}}$，而第二个方程组的解为 $X=(10,-2)^{\mathrm{T}}$.

这个例子说明，即使两个线性方程组的系数矩阵及右端项变

化很小,但它们的解却相差很大. 如果把其中的一个方程组看成另一个方程组的近似方程组,那么,即使把近似方程组解得再精确,也没有什么意义.

以上这种研究方程组的系数矩阵(或右端项)有微小误差(或称有一小的扰动)对方程组的解向量的影响问题,就称为线性代数方程组的解关于系数的敏感性问题,这样一种分析方法也常称为"扰动分析". 从下面的讨论,我们将看到,这种敏感性与系数矩阵 A 有密切关系,也即取决于方程组的固有性质——方程组的性态. 并可通过进一步分析,引进刻划方阵性态的量——条件数的概念. 这就要用到向量和矩阵的范数知识.

设求解方程组 $AX=b$ 的扰动方程组为

$$(A+\delta A)(X+\delta X)=b+\delta b$$

其中,δA 叫做 A 的扰动矩阵,δX 和 δb 分别叫做 X 和 b 的扰动向量.

以下总假定 $A+\delta A$ 为非奇异.

为了讨论简单起见,下面对扰动 δb 和 δA 分两种情况讨论:

(1) $\delta A=0$ 的情况

设方程组 $AX=b$ 中,系数矩阵 A 非奇异且是精确的,即没有扰动. 右端向量 $b\neq 0$,且有扰动 δb(它是由各右端项的误差组成的向量),并假设计算过程没有引入其他误差(即每步计算都是准确的),由 δb 引起的解的扰动记为 δX,问此时解向量的相对扰动 $\|\delta X\|/\|X\|$ 有多大?

显然,计算解 $X+\delta X$ 应满足方程组

$$A(X+\delta X)=(b+\delta b) \tag{5.36}$$

用式(5.36)减去 $AX=b$ 得

$$A\delta X=\delta b \tag{5.37}$$

由式(5.37)有

或

$$\delta \boldsymbol{X}=\boldsymbol{A}^{-1}\delta \boldsymbol{b}$$

从而,由范数的相容性得不等式

$$\|\delta \boldsymbol{X}\| \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{b}\|$$

再由 $\boldsymbol{AX}=\boldsymbol{b}$,又有不等式

$$\|\boldsymbol{b}\| \leqslant \|\boldsymbol{A}\| \cdot \|\boldsymbol{X}\|$$

从而有

$$\|\delta \boldsymbol{X}\| \cdot \|\boldsymbol{b}\| \leqslant \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{X}\| \cdot \|\delta \boldsymbol{b}\| \tag{5.38}$$

因 $\boldsymbol{b}\neq 0$,所以 $\boldsymbol{X}\neq 0$,则有 $\|\boldsymbol{b}\|>0$, $\|\boldsymbol{X}\|>0$,

由式(5.38)可得

$$\frac{\|\delta \boldsymbol{X}\|}{\|\boldsymbol{X}\|} \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\| \cdot \frac{\|\delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|} \tag{5.39}$$

式(5.39)给出了解向量的精度的一种估计,它表明解的相对误差有可能放大原始数据(右端项)相对误差的 $\|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|$ 倍.

(2) $\delta \boldsymbol{b}=0$ 的情况

现设方程组右端无扰动,而系数矩阵 A 有扰动 δA,于是方程组

$$(\boldsymbol{A}+\delta \boldsymbol{A})\widetilde{\boldsymbol{X}}=\boldsymbol{b} \text{ 有解为 } \widetilde{\boldsymbol{X}}=\boldsymbol{X}+\delta \boldsymbol{X}$$

即

$$(\boldsymbol{A}+\delta \boldsymbol{A}) \cdot (\boldsymbol{X}+\delta \boldsymbol{X})=\boldsymbol{b} \tag{5.40}$$

或

$$\delta \boldsymbol{A}(\boldsymbol{X}+\delta \boldsymbol{X})+\boldsymbol{A}\delta \boldsymbol{X}=0$$

则

$$\delta \boldsymbol{X}=-\boldsymbol{A}^{-1} \cdot \delta \boldsymbol{A}(\boldsymbol{X}+\delta \boldsymbol{X})$$

有

$$\|\delta \boldsymbol{X}\| \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}\| \cdot \|\boldsymbol{X}+\delta \boldsymbol{X}\|$$

若设 $\boldsymbol{A}+\delta \boldsymbol{A}$ 可逆且 $\boldsymbol{b}\neq 0$,从而 $\boldsymbol{X}+\delta \boldsymbol{X}\neq 0$,

故得
$$\frac{\|\delta \boldsymbol{X}\|}{\|\boldsymbol{X}+\delta \boldsymbol{X}\|} \leqslant \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|} \tag{5.41}$$

公式(5.41)同样表明,当 A 有扰动 δA 时,所引起的解 $\boldsymbol{X}+\delta \boldsymbol{X}$ 的相对误差仍不超过 A 的相对误差的一个倍数.无论是式(5.39)还是式(5.41),起控制作用的数 $\|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|$ 反映 A 和方程组本身的性态,我们给量 $\|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|$ 一个名称.

定义 5.10 设 A 为非奇异矩阵,称数
$$\mathrm{Cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \tag{5.42}$$
为矩阵 A 的条件数.

由上述讨论知,条件数小,扰动所引起的解的相对误差一定小;条件数大,扰动所引起的相对误差就可能很大.当方程组的系数矩阵和右端项同时有扰动时,这个结论仍然成立.一般情况下,在 $\|\boldsymbol{A}^{-1} \cdot \delta \boldsymbol{A}\| \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}\| < 1$ 的条件下可推得
$$\frac{\|\delta \boldsymbol{X}\|}{\|\boldsymbol{X}\|} \leqslant \frac{\mathrm{Cond}(\boldsymbol{A})}{1-\mathrm{Cond}(\boldsymbol{A})\dfrac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|}}\left(\frac{\|\delta b\|}{\|b\|}+\frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|}\right).$$

条件数与所取的矩阵范数有关,最常用的是 $\|\cdot\|_\infty$ 和 $\|\cdot\|_2$.

由于 $\mathrm{Cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \geqslant \|\boldsymbol{A}\boldsymbol{A}^{-1}\| = \|\boldsymbol{I}\| = 1$

可见,条件数是一个放大的倍数,且总是以 1 为下界.

对一个确定的线性代数方程组,若系数矩阵的条件数相对地小,我们就称这个方程组是良态的;反之,条件数相对地大,就称方程组是病态的.同时把 A 称为(对解方程组而言)良态阵或病态阵.用一个稳定的方法去解一个良态的方程组,必然得到较精确的结果.同样,用一个稳定的方法去解一个病态的方程组,结果就可能很差.

当 A 为正交阵时,有 $\mathrm{Cond}(\boldsymbol{A})_2 = 1$,所以系数矩阵为正交阵的方程组是良态的.

希尔伯特(Hilbert)矩阵

$$H_n=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n}\\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1}\end{bmatrix}$$

是很有名的病态阵.例如,当 $n=3$ 时,

$$\|H_3\|_\infty=\frac{11}{6},\ \|H_3^{-1}\|_\infty=408,\mathrm{Cond}_\infty(H_3)=748.$$

当 $n=6$ 时,$\mathrm{Cond}_\infty(H_6)\approx 29\times 10^6$.随着 n 的增大,条件数增长很快,从而希尔伯特矩阵病态愈严重.

由上述讨论知,要判别一个矩阵是否病态,需要计算矩阵的条件数 $\mathrm{Cond}(A)=\|A\|\cdot\|A^{-1}\|$,而条件数的计算首先要计算逆阵的范数,这个计算工作量非常大,在实际使用时很不方便.但一般如下的一些现象可作为判断病态阵的参考.

① 若用主元消去法过程中出现小主元;

② 系数矩阵行列式的绝对值相对来说很小;

③ 从矩阵本身来看,若元素间数量级相差很大且无一定规律,或者矩阵的某些行(列)近似线性相关,这样的矩阵则有可能是病态阵.

对于病态方程组的计算要十分小心,一般可采用高精度算法、预处理方法、平衡方法和迭代改善法等.(可参考[1])

5.5 迭代法

前面几节,我们讨论了解线性代数方程组的直接法,它对于阶

数不高的中小型方程组是很有效的，但对于阶数较高的又不具有带状结构的线性代数方程组，则常用迭代法.本节介绍解线性代数方程组的另一大类方法——迭代法.

迭代法的基本思想是用逐次逼近的方法去求线性代数方程组的解.

设方程组

$$\boldsymbol{AX}=\boldsymbol{b}$$

的系数矩阵 $\boldsymbol{A}$ 非奇异，从而有且仅有一组解.

用迭代法求解 $\boldsymbol{AX}=\boldsymbol{b}$，先把方程组化为等价的便于迭代的形式 $\boldsymbol{X}=\boldsymbol{BX}+\boldsymbol{g}$，这一点总是能办到的，如，可令 $\boldsymbol{A}=\boldsymbol{I}-\boldsymbol{B}$，即得上式，再写成迭代格式为

$$\boldsymbol{X}^{(m+1)}=\boldsymbol{BX}^{(m)}+\boldsymbol{g} \quad ,(m=0,1,\cdots) \tag{5.43}$$

向量 $\boldsymbol{X}^{(0)}$ 事先给定，称为初始向量，用公式(5.43)逐步迭代求解的方法叫做迭代法，也称为一阶线性定常迭代法.

如果由式(5.43)形成的序列$\{\boldsymbol{X}^{(m)}\}$，有

$$\lim_{m\to\infty}\boldsymbol{X}^{(m)}=\boldsymbol{X}^{*}$$

则称迭代法式(5.43)是收敛的，否则称迭代法发散.

由于迭代法不需要存贮系数矩阵的零元素，从而迭代法适合于求解系数矩阵为大型(高阶)稀疏矩阵的方程组，同时计算机程序比较简单，这些都是迭代法的优点.生成向量序列$\{\boldsymbol{X}^{(m)}\}$的规则不同，就形成不同的迭代法.本节重点介绍三种迭代法，即雅可比(Jacobi)迭代法、赛德尔(Seidel)迭代法以及超松弛迭代法(SOR 法).

迭代法一个突出的问题是收敛性问题，因此，本节还要介绍一些有关收敛性讨论的若干重要结论，给出几个常用的收敛的判别法.

一、雅可比(Jacobi)迭代法

雅可比迭代法是一种最简单的迭代法,所以也称为简单迭代法.

设 n 阶线性代数方程组

$$\begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=b_n \end{cases} \tag{5.44}$$

的系数矩阵 $\boldsymbol{A}$ 非奇异,且对角元 $a_{11},a_{22},\cdots,a_{nn}$ 均不为零.则可分别从方程组(5.44)的第 i 个方程解出 $x_i(i=1,2,\cdots,n)$.这样,方程组(5.44)就改写为等价的方程组

$$\begin{cases} x_1=\dfrac{1}{a_{11}}(b_1-a_{12}x_2-\cdots-a_{1n}x_n) \\ x_2=\dfrac{1}{a_{22}}(b_2-a_{21}x_1-\cdots-a_{2n}x_n) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n=\dfrac{1}{a_{nn}}(b_n-a_{n1}x_1-\cdots-a_{n,n-1}x_{n-1}) \end{cases} \tag{5.45}$$

选取初始向量 $\boldsymbol{X}^{(0)}=(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)})^{\mathrm{T}}$,将其代入方程组(5.45)的右端,进行第一次迭代,计算结果记为

$$\boldsymbol{X}^{(1)}=(x_1^{(1)},x_2^{(1)},\cdots,x_n^{(1)})^{\mathrm{T}}$$

再将 $\boldsymbol{X}^{(1)}$ 代入方程组(5.45)式右端,进行第 2 次迭代,计算结果记为 $\boldsymbol{X}^{(2)}=(x_1^{(2)},x_2^{(2)},\cdots,x_n^{(2)})^{\mathrm{T}}$,如此继续,就得到了一般迭代格式如下:

$$
\begin{cases}
x_1^{(m+1)} = \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(m)} - a_{13}x_3^{(m)} - \cdots - a_{1n}x_n^{(m)}) \\
x_2^{(m+1)} = \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(m)} - a_{23}x_3^{(m)} - \cdots - a_{2n}x_n^{(m)}) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n^{(m+1)} = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(m)} - a_{n2}x_2^{(m)} - \cdots - a_{n,n-1}x_{n-1}^{(m)})
\end{cases}
\tag{5.46}
$$

$$(m=0,1,2,\cdots)$$

按照式(5.46)**进行迭代得出向量序列**$\{\boldsymbol{X}^{(m)}\}$**的方法就叫做解线性代数方程组** $\boldsymbol{AX}=\boldsymbol{b}$ **的雅可比迭代法，简称为 J-迭代法. 其迭代公式的分量形式可简记为**

$$
\begin{cases}
x_i^{(m+1)} = \frac{1}{a_{ii}}(b_i - \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^{(m)}) \\
\text{其中 } \boldsymbol{X}^{(m)} = (x_1^{(m)}, x_2^{(m)}, \cdots, x_n^{(m)})^{\mathrm{T}}
\end{cases}
\tag{5.47}
$$

$$(m=0,1,2,\cdots,i=1,2,\cdots)$$

由此可见，雅可比迭代法的迭代公式是如此简单，每迭代一次只需计算一次矩阵和向量的乘法，在计算机上计算时需要用两组工作单元，以便存贮 $\boldsymbol{X}^{(m)}$ **和** $\boldsymbol{X}^{(m+1)}$**. 但必须注意，上述所得的向量序列**$\{\boldsymbol{X}^{(m)}\}$**是否必然收敛于** $\boldsymbol{AX}=\boldsymbol{b}$ **的解是有条件的，且若同样是收敛的格式，还有收敛速度快、慢的问题.**

二、赛德尔迭代法

赛德尔迭代法也称高斯-赛德尔(Gauss-Seidel)**迭代法，简记为** G-S **方法，它是雅可比方法的改进.**

分析一下 J-**迭代的计算过程，若向量序列**$\{\boldsymbol{X}^{(m)}\}$**收敛于方程组的精确解，则显然** $\boldsymbol{X}^{(m+1)}$ **比** $\boldsymbol{X}^{(m)}$ **应更靠近精确解.**

事实上，由雅可比方法的迭代公式知，迭代过程的每一步计算

是用 $\boldsymbol{X}^{(m)}$ 的全部分量来计算 $\boldsymbol{X}^{(m+1)}$ 的所有分量. 显然,在计算 $\boldsymbol{X}^{(m+1)}$ 的第 i 个分量 $x_i^{(m+1)}(i>1)$ 时,对已经计算出的最新的分量 $x_1^{(m+1)},x_2^{(m+1)},\cdots,x_{i-1}^{(m+1)}$ 没有利用,而这些最新计算出来的分量可能比旧的 $\boldsymbol{X}^{(m)}$ 的分量更近于精确解,因此,对这些最新分量 $x_j^{(m+1)}$ $(j=1,2,\cdots,i-1)$ 加以利用是合理的. 如果逐次改用 $x_j^{(m+1)}$ 代替 J-迭代中的 $x_j^{(m)}(j=1,2,\cdots,i-1)$,则就得到所谓解方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的高斯-赛德尔迭代法.

G-S 迭代法的迭代公式为

$$\begin{cases} x_1^{(m+1)}=\dfrac{1}{a_{11}}(b_1-a_{12}x_2^{(m)}-a_{13}x_3^{(m)}-\cdots-a_{1n}x_n^{(m)}) \\ x_2^{(m+1)}=\dfrac{1}{a_{22}}(b_2-a_{21}x_1^{(m+1)}-a_{23}x_3^{(m)}-\cdots-a_{2n}x_n^{(m)}) \\ \qquad\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(m+1)}=\dfrac{1}{a_{nn}}(b_n-a_{n1}x_1^{(m+1)}-a_{n2}x_2^{(m+1)}-\cdots-a_{n,n-1}x_{n-1}^{(m+1)}) \end{cases} \tag{5.48}$$

其分量形式可统一记为

$$\begin{cases} x_i^{(m+1)}=\dfrac{1}{a_{ii}}\{b_i-\sum\limits_{j=1}^{i-1}a_{ij}x_j^{(m+1)}-\sum\limits_{j=i+1}^{n}a_{ij}x_j^{(m)}\} \\ \qquad\quad =x_i^{(m)}+\dfrac{1}{a_{ii}}\{b_i-\sum\limits_{j=1}^{i-1}a_{ij}x_j^{(m+1)}-\sum\limits_{j=i}^{n}a_{ij}x_j^{(m)}\} \\ \text{其中 } \boldsymbol{X}^{(m)}=(x_1^{(m)},x_2^{(m)},\cdots,x_n^{(m)})^{\mathrm{T}} \end{cases} \tag{5.49}$$

$$(m=0,1,2,\cdots;i=1,2,\cdots,n)$$

由此可见,赛德尔迭代法的特点是:逐次用计算出来的新值代替旧值. 因为一旦求得 $x_j^{(m+1)}$ 后,就不需要再保存 $x_j^{(m)}$ 了$(1\leqslant j\leqslant n)$,因而在计算机上计算时只需要一组工作单元,这是赛德尔迭代法比较 J-迭代的一个明显的优点,且在收敛的情况下,赛德尔迭代法收敛的速度要比雅可比迭代法快.

三、迭代公式的矩阵形式

为了便于讨论迭代法的收敛性，下面用矩阵来描述迭代公式.

首先，将方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的系数方阵 $\boldsymbol{A}$ 拆成

$$\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$$

其中，$\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$，是由 $\boldsymbol{A}$ 的对角元组成的对角矩阵，$-\boldsymbol{L}$ 与 $-\boldsymbol{U}$ 分别是 $\boldsymbol{A}$ 的对角线下方元素和上方元素组成的严格下三角矩阵与严格上三角矩阵.

也即

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}$$

$$-\begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & -a_{n,n-1} & & 0 \end{bmatrix}$$

$$-\begin{bmatrix} 0 & -a_{12} & \cdots & \cdots & -a_{1n} \\ & 0 & -a_{23} & \cdots & -a_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & -a_{n-1,n} \\ & & & & 0 \end{bmatrix}$$

则雅可比迭代法的矩阵形式为

$$\boldsymbol{X}^{(m+1)}=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{X}^{(m)}+\boldsymbol{D}^{-1}\boldsymbol{b} \tag{5.50}$$

对于 G-S 迭代法，也可写为矩阵形式

$$DX^{(m+1)}=LX^{(m+1)}+UX^{(m)}+b$$

于是有

$$(D-L)X^{(m+1)}=UX^{(m)}+b$$

由设 $a_{ii}\neq 0,(i=1,2,\cdots,n)$,所以 $D-L$ 非奇异.

从而有

$$X^{(m+1)}=(D-L)^{-1}UX^{(m)}+(D-L)^{-1}b \tag{5.51}$$

这就是 G-S 迭代法常用的矩阵形式.

我们也可以把式(5.50)与式(5.51)统一地写成

$$X^{(m+1)}=BX^{(m)}+g \tag{5.52}$$

对于 J-迭代法来说,其中 B 和 g 分别为

$$B_{\mathrm{J}}=D^{-1}(L+U),\quad g_{\mathrm{J}}=D^{-1}b$$

对于 G-S 迭代法来说,其中 B 和 g 分别为

$$B_{\mathrm{S}}=(D-L)^{-1}U,\quad g_{\mathrm{S}}=(D-L)^{-1}b$$

不妨把式(5.52)叫做一般迭代过程.

下面我们先就一般迭代过程:$X^{(m+1)}=BX^{(m)}+g$ 来研究其收敛性.

四、一般迭代法的收敛性

将方程组 $AX=b$,改写为与之等价的方程组

$$X=BX+g \tag{5.53}$$

其迭代形式为

$$X^{(m+1)}=BX^{(m)}+g \tag{5.54}$$

B 称为迭代矩阵,g 称为右端向量.

显然,若由式(5.54)生成的向量序列$\{X^{(m)}\}$收敛,即存在极限

$$\lim_{m\to\infty}X^{(m)}=X^{*}$$

则由式(5.54),必有

$$\boldsymbol{X}^{*}=\boldsymbol{B}\boldsymbol{X}^{*}+\boldsymbol{g}$$

也就是说,$\{\boldsymbol{X}^{(m)}\}$必收敛于式(5.53)的解.

定理 5.9 (迭代法基本定理)

设有方程组 $\boldsymbol{X}=\boldsymbol{B}\boldsymbol{X}+\boldsymbol{g}$

对于任意初始向量 $\boldsymbol{X}^{(0)}$ 及右端向量 $\boldsymbol{g}$,解此方程组的一般迭代法收敛的充要条件是迭代矩阵 $\boldsymbol{B}$ 的谱半径

$$\rho(\boldsymbol{B})<1$$

(证明从略)

并可以证明,$\rho(\boldsymbol{B})$愈小,收敛速度愈快.且从上面的讨论已知一般迭代法 $\boldsymbol{X}^{(m+1)}=\boldsymbol{B}\boldsymbol{X}^{(m)}+\boldsymbol{g}$ 是否收敛,收敛速度的快慢,只取决于迭代矩阵,而与右端向量 $\boldsymbol{g}$ 及如何选取迭代的初始向量无关.但是一般而言,计算矩阵的谱半径较困难(特别是在方阵的阶数 n 较大时),下面给出用矩阵范数判定收敛的定理.

为此,先给出一个引理:

引理 若 n 阶方阵 $\boldsymbol{B}$ 满足 $\|\boldsymbol{B}\|<1$,则

(1) $\boldsymbol{I}\pm\boldsymbol{B}$ 为非奇异矩阵;

$$(2)\quad \|(\boldsymbol{I}\pm\boldsymbol{B})^{-1}\|\leqslant\frac{1}{1-\|\boldsymbol{B}\|} \tag{5.55}$$

定理 5.10 (迭代法收敛的充分条件)

设一般迭代法 $\boldsymbol{X}^{(m+1)}=\boldsymbol{B}\boldsymbol{X}^{(m)}+\boldsymbol{g}$ 的迭代矩阵 $\boldsymbol{B}$ 满足

$$\|\boldsymbol{B}\|=q<1$$

则 (1) 对于任意的初始向量 $\boldsymbol{X}^{(0)}$ 与右端向量 $\boldsymbol{g}$,迭代法收敛于方程组的精确解 $\boldsymbol{X}^{*}$.

且有如下二个误差估计式:

(2) $\| \boldsymbol{X}^{(m)} - \boldsymbol{X}^{*} \| \leqslant \frac{q^{m}}{1-q} \| \boldsymbol{X}^{(1)} - \boldsymbol{X}^{(0)} \|$ (5.56)

(3) $\| \boldsymbol{X}^{(m)} - \boldsymbol{X}^{*} \| \leqslant \frac{q}{1-q} \| \boldsymbol{X}^{(m)} - \boldsymbol{X}^{(m-1)} \|$ (5.57)

证明 (1) 由于方阵 $\boldsymbol{B}$ 的谱半径不超过 $\boldsymbol{B}$ 的任何一种范数.即

$$\rho(\boldsymbol{B}) \leqslant \| \boldsymbol{B} \|$$

则当 $\| \boldsymbol{B} \| = q < 1$ 时,有 $\rho(\boldsymbol{B}) < 1$,由定理 5.9 知一般迭代法收敛.

(2)(不证明)为了使误差向量 $\| \boldsymbol{X}^{(m)} - \boldsymbol{X}^{*} \|$ 小于要求的精度 ε,可以利用这一估计式来估计需要迭代的次数,但一般说来,这样计算出来的迭代次数都偏大,在实际问题中很少采用,一般用估计式(5.57).

(3) 由于

$$\varepsilon_m = \boldsymbol{X}^{(m)} - \boldsymbol{X}^{*} = (\boldsymbol{B}\boldsymbol{X}^{(m-1)} + \boldsymbol{g}) - (\boldsymbol{B}\boldsymbol{X}^{*} + \boldsymbol{g}) = \boldsymbol{B}\boldsymbol{X}^{(m-1)} - \boldsymbol{B}\boldsymbol{X}^{*}$$

而

$$\boldsymbol{X}^{*} = \boldsymbol{B}\boldsymbol{X}^{*} + \boldsymbol{g}$$

得

$$(\boldsymbol{I} - \boldsymbol{B})\boldsymbol{X}^{*} = \boldsymbol{g}$$

因为 $\| \boldsymbol{B} \| < 1$,则 $\boldsymbol{I} - \boldsymbol{B}$ 非奇异,则 $(\boldsymbol{I} - \boldsymbol{B})^{-1}$ 存在,所以

$$X^{*} = (\boldsymbol{I} - \boldsymbol{B})^{-1} \boldsymbol{g}$$

则

$$\varepsilon_m = \boldsymbol{B}\boldsymbol{X}^{(m-1)} - \boldsymbol{B}(\boldsymbol{I} - \boldsymbol{B})^{-1} \boldsymbol{g}$$

$$= \boldsymbol{B}(\boldsymbol{I} - \boldsymbol{B})^{-1} [(\boldsymbol{I} - \boldsymbol{B})\boldsymbol{X}^{(m-1)} - \boldsymbol{g}]$$

(因 $\boldsymbol{g} = \boldsymbol{X}^{(m)} - \boldsymbol{B}\boldsymbol{X}^{(m-1)}$)

$$= \boldsymbol{B}(\boldsymbol{I} - \boldsymbol{B})^{-1} [X^{(m-1)} - \boldsymbol{X}^{(m)}]$$

两端取范数,由 $\| (\boldsymbol{I} - \boldsymbol{B})^{-1} \| \leqslant \frac{1}{1 - \| \boldsymbol{B} \|}$

得 $$\|\boldsymbol{X}^{(m)}-\boldsymbol{X}^{*}\|\leqslant\frac{\|\boldsymbol{B}\|}{1-\|\boldsymbol{B}\|}\|\boldsymbol{X}^{(m)}-\boldsymbol{X}^{(m-1)}\|.$$

式(5.57)告诉我们,只要 $\|\boldsymbol{B}\|$ 不是很接近于 1,则当相邻两次迭代向量 $\boldsymbol{X}^{(m)}$ 和 $\boldsymbol{X}^{(m-1)}$ 很接近时,$\boldsymbol{X}^{(m)}$ 与 $\boldsymbol{X}^{*}$ 也很接近,因此可以用 $\|\boldsymbol{X}^{(m)}-\boldsymbol{X}^{(m-1)}\|<\varepsilon$ 作为迭代终止的条件. 并可将 $\boldsymbol{X}^{(m)}$ 作为方程组的近似解.

由上述定理可知,$\|\boldsymbol{B}\|=q<1$ 愈小,迭代法收敛愈快.

由于 $\|\boldsymbol{B}\|_1$, $\|\boldsymbol{B}\|_\infty$ 都能很方便地用 $\boldsymbol{B}$ 的元素来表示,因此常用

① $\|\boldsymbol{B}\|_\infty=\max\limits_{i}\sum\limits_{j=1}^{n}|b_{ij}|<1$

或 ② $\|\boldsymbol{B}\|_1=\max\limits_{j}\sum\limits_{i=1}^{n}|b_{ij}|<1$

来作为判别一般迭代法是否收敛的充分条件:当 $\|\boldsymbol{B}\|<1$ 时,迭代必然收敛,不过要注意,$\|\boldsymbol{B}\|<1$ 仅是收敛的充分条件而不是必要条件.

以上讨论了一般迭代法的收敛性问题,这对于雅可比迭代法和赛德尔迭代法当然也适用,但由于它们的特殊性,下面我们给出直接由系数矩阵 A 的特性来判别雅可比迭代法和赛德尔迭代法收敛的充分条件,这种判别法不要求写成具体的迭代格式 $\boldsymbol{X}^{(m+1)}=\boldsymbol{B}\boldsymbol{X}^{(m)}+\boldsymbol{g}$,而由原方程组 $\boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}$ 的系数矩阵 A 即可判定,所以应用时更方便.

五、雅可比迭代法和赛德尔迭代法收敛的充分条件

先介绍几个重要概念:

1. 矩阵的对角占优

定义 5.11 设 $\boldsymbol{A}=(a_{ij})_{n\times n}\in\boldsymbol{R}^{n\times n}$,如果对于一切 $i(1\leqslant i\leqslant n)$

都有

$$|a_{ii}| \geqslant \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \tag{5.58}$$

且至少有一个 i,例如 $i=i_0$,使式(5.58)成为真不等式,即

$$|a_{i_0 i_0}| > \sum_{\substack{j=1 \\ j \neq i_0}}^{n} |a_{i_0 j}| \text{成立}$$

则称 $\boldsymbol{A}$ 为按行弱对角占优矩阵.

若对一切 $i(1 \leqslant i \leqslant n)$ 均有不等式

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \tag{5.59}$$

成立,则称 $\boldsymbol{A}$ 为按行严格对角占优矩阵(即 $\boldsymbol{A}$ 的每一行对角元素的绝对值都严格大于同行其他元素绝对值之和).

例 5.8

$$\boldsymbol{A}=\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \quad \boldsymbol{B}=\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

$\boldsymbol{A}$ 是按行弱对角占优矩阵,$\boldsymbol{B}$ 是按行严格对角占优矩阵.

类似地可定义按列为严格对角占优与弱对角占优的矩阵.

2. 矩阵的可约与不可约

定义 5.12 设 $\boldsymbol{A}=(a_{ij})_{n\times n} \in \boldsymbol{R}^{n\times n}(n \geqslant 2)$,如果 $\boldsymbol{A}$ 可通过“对称的”行的次序调换和相应的列的次序的调换而成为

$$\widetilde{\boldsymbol{A}}=\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{bmatrix} \tag{5.60}$$

其中,$\boldsymbol{A}_{11}$ 为 r 阶方阵,$\boldsymbol{A}_{22}$ 为 $n-r$ 阶方阵$(1 \leqslant r < n)$,$\boldsymbol{O}$ 为零阵,则

称 A 为可约矩阵;否则,则称 A 为不可约矩阵.

注意,以上行与行以及列与列的互换是“对称的”,也即每作一个行互换就要相应地作一个列互换,且互换的行序号和互换的列的序号是一样的.

例如

$$\boldsymbol{A}=\begin{bmatrix}9 & 1 & -2 & 1 & 1\\0 & 10 & 0 & 1 & 0\\2 & 1 & 16 & -1 & -3\\0 & -1 & 0 & 6 & 0\\4 & 0 & 1 & -1 & 23\end{bmatrix}\xrightarrow{(2),(5)\text{二行交换}}$$

$$\longrightarrow\begin{bmatrix}9 & 1 & -2 & 1 & 1\\4 & 0 & 1 & -1 & 23\\2 & 1 & 16 & -1 & -3\\0 & -1 & 0 & 6 & 0\\0 & 10 & 0 & 1 & 0\end{bmatrix}$$

$$\xrightarrow{(2),(5)\text{二列交换}}\left[\begin{array}{ccc:cc}9 & 1 & -2 & 1 & 1\\4 & 23 & 1 & -1 & 0\\2 & -3 & 16 & -1 & 1\\\hdashline 0 & 0 & 0 & 6 & -1\\0 & 0 & 0 & 1 & 10\end{array}\right]$$

所以 A 是可约的.

显然,若 A 中的元素全部为非零元素,则 A 肯定是不可约的;若 A 不可约,对于 $\mu\neq0$(常数),则 μA 也不可约. 三对角阵 $A=[a_i,b_i,c_i]_1^n$,当 a_i,b_i,c_i 均不为零时也是不可约的. 由上述定义容易证明:

定理 5.11 若 $A\in\boldsymbol{R}^{n\times n}$ 按行(或按列)为严格对角占优阵或为不可约弱对角占优阵,则 A 为非奇异矩阵.(证略)

下面给出直接根据给定方程组的系数矩阵 A 判断雅可比迭

代与赛德尔迭代收敛的三个充分判别条件.

［判别条件Ⅰ］ 若线性代数方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的系数方阵 $\boldsymbol{A}=[a_{ij}]_{n\times n}\in\boldsymbol{R}^{n\times n}$ 满足下列条件之一：

① 按行(或按列)为严格对角占优；

② 不可约且按行(或按列)为弱对角占优；

则雅可比迭代法和赛德尔迭代法都是收敛的.

证明 ① 只证 $\boldsymbol{A}$ 为按行严格对角占优情况下,雅可比迭代法收敛.

由于 $\boldsymbol{A}$ 为按行严格对角占优,则对一切 $i(1\leqslant i\leqslant n)$,均有

$$|a_{ii}|>\sum_{j\neq i}|a_{ij}|,\quad \text{所以}\quad \sum_{j\neq i}\left|\frac{a_{ij}}{a_{ii}}\right|<1$$

而雅可比迭代法的迭代矩阵

$$\boldsymbol{B}_{\mathrm{J}}=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})=\begin{bmatrix} 0 & -\dfrac{a_{12}}{a_{11}} & \cdots & -\dfrac{a_{1n}}{a_{11}} \\ -\dfrac{a_{21}}{a_{22}} & 0 & \cdots & -\dfrac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ -\dfrac{a_{n1}}{a_{nn}} & -\dfrac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix}$$

满足

$$\|\boldsymbol{B}_{\mathrm{J}}\|_{\infty}=\max_{i}\sum_{\substack{j=1\\ j\neq i}}^{n}\left|\frac{a_{ij}}{a_{ii}}\right|<1.$$

由定理 5.10 知雅可比迭代法收敛.

② 只证 $\boldsymbol{A}$ 为不可约且按行为弱对角占优的情况下,赛德尔迭代法收敛. 根据定理 5.9,只要证明赛德尔迭代法的迭代矩阵 $\boldsymbol{B}_{\mathrm{S}}$ 的谱半径 $\rho(\boldsymbol{B}_{\mathrm{S}})<1$.

(用反证法)在 $\boldsymbol{A}$ 为不可约且按行为弱对角占优的情况下,若赛德尔迭代法的迭代矩阵 $\boldsymbol{B}_{\mathrm{S}}$ 有某一特征值 μ 满足 $|\mu|\geqslant 1$,则一方面,因为 μ 是 $\boldsymbol{B}_{\mathrm{S}}$ 的特征值,应有 $\det(\mu\boldsymbol{I}-\boldsymbol{B}_{\mathrm{S}})=0$,而由

$$\boldsymbol{B}_S=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}$$

有 $\mu\boldsymbol{I}-\boldsymbol{B}_S=\mu\boldsymbol{I}-(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}=(\boldsymbol{D}-\boldsymbol{L})^{-1}[(\boldsymbol{D}-\boldsymbol{L})\mu-\boldsymbol{U}]$

所以 $\det(\mu\boldsymbol{I}-\boldsymbol{B}_S)=\det(\boldsymbol{D}-\boldsymbol{L})^{-1}\det[\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U}]=0$

由于 $$\det((\boldsymbol{D}-\boldsymbol{L})^{-1})\neq 0$$

所以 $$\det(\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U})=0$$

另一方面,矩阵

$$\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U}=\begin{bmatrix} \mu a_{11} & a_{12} & & \cdots & & a_{1n} \\ \mu a_{21} & \mu a_{22} & & \cdots & & a_{2n} \\ \vdots & \vdots & \ddots & & & \vdots \\ \mu a_{i1} & \mu a_{i2} & \cdots & \mu a_{ij} & & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu a_{n1} & \mu a_{n2} & \cdots & \mu a_{nj} & \cdots & \mu a_{nn} \end{bmatrix}$$

是 A 的包括主对角线在内的下三角部分的每个元素都乘上 μ $(|\mu|\geqslant 1)$.

由于 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$ 是不可约的,而 $\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U}$ 的非零元素与 $\boldsymbol{A}$ 的完全相同. 所以 $\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U}$ 也是不可约的. 其次,由于 $|\mu|\geqslant 1$ 和已知 $\boldsymbol{A}$ 为弱对角占优,所以

$$|\mu a_{ii}|=|\mu||a_{ii}|\geqslant|\mu|\sum_{j\neq i}|a_{ij}|\geqslant\sum_{j=1}^{i-1}|\mu a_{ij}|+\sum_{j=i+1}^{n}|a_{ij}|$$

且至少有一 $i=i_0$,使不等式

$$|\mu a_{i_0 i_0}|>\sum_{j=1}^{i_0-1}|\mu a_{i_0 j}|+\sum_{j=i_0+1}^{n}|a_{i_0 j}|$$

成立. 这就是说,矩阵 $\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U}$ 也是弱对角占优的.

综上所述,$\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U}$ 是不可约且弱对角占优的.

根据定理 5.11, $\det(\mu\boldsymbol{D}-\mu\boldsymbol{L}-\boldsymbol{U})\neq 0$. 这和前面已证的

$\det(\mu \boldsymbol{D}-\mu \boldsymbol{L}-\boldsymbol{U})=0$矛盾. 这矛盾说明不可假定$|\mu|\geqslant 1$,因此,只能是$|\mu|<1$,从而$\rho(\boldsymbol{B}_S)<1$. 所以赛德尔迭代法收敛.

用有关的代数知识和迭代法收敛定理还可证明:

[判别条件Ⅱ] 若线性代数方程组$\boldsymbol{AX}=\boldsymbol{b}$的系数矩阵$\boldsymbol{A}$为对称正定阵,则赛德尔迭代法收敛.

[判别条件Ⅲ] 若线性代数方程组$\boldsymbol{AX}=\boldsymbol{b}$的系数方阵$\boldsymbol{A}$为对称正定阵,$2\boldsymbol{D}-\boldsymbol{A}$也为对称正定阵,则雅可比迭代法收敛;若$\boldsymbol{A}$为对称正定阵而$2\boldsymbol{D}-\boldsymbol{A}$为非正定阵,则雅可比迭代法不收敛(其中$\boldsymbol{D}$为$\boldsymbol{A}$的对角元组成的对角阵,所以$2\boldsymbol{D}-\boldsymbol{A}$与$\boldsymbol{A}$只是非对角元的符号不同).

例 5.9 设在$\boldsymbol{AX}=\boldsymbol{b}$中

$$\boldsymbol{A}=\begin{bmatrix}1 & 0.9 & 0.9\\ 0.9 & 1 & 0.9\\ 0.9 & 0.9 & 1\end{bmatrix}$$

因为$\boldsymbol{A}$对称且各阶主子式皆大于零,所以$\boldsymbol{A}$为对称正定阵,由[判别条件Ⅱ]知用赛德尔迭代法收敛.

但在

$$2\boldsymbol{D}-\boldsymbol{A}=\begin{bmatrix}1 & -0.9 & -0.9\\ -0.9 & 1 & -0.9\\ -0.9 & -0.9 & 1\end{bmatrix}$$

中

$\det(2\boldsymbol{D}-\boldsymbol{A})<0$, $2\boldsymbol{D}-\boldsymbol{A}$是非正定阵,按[判别条件Ⅲ],用雅可比迭代法不收敛.

在解线性代数方程组时,若系数矩阵不满足迭代法收敛条件,有时只要适当调换方程的次序,就能得到满足收敛条件的系数阵.

例 5.10 设方程组

$$\begin{cases}3x_1-10x_2=-7\\ 9x_1-4x_2=5\end{cases}$$

对系数矩阵 $A=\begin{bmatrix}3 & -10\\ 9 & -4\end{bmatrix}$ 使用上述判别条件，皆无法判断，而由计算得

$$\rho(\boldsymbol{B}_{\mathrm{J}})=\frac{\sqrt{30}}{2}>1,\quad \rho(\boldsymbol{B}_{\mathrm{S}})=\frac{15}{2}>1$$

所以雅可比迭代法和赛德尔迭代法皆不收敛.

但只要交换两个方程的次序，得等价的方程组为

$$\begin{cases}9x_1-4x_2=5\\ 3x_1-10x_2=-7\end{cases}$$

可知 $\widetilde{\boldsymbol{A}}=\begin{pmatrix}9 & -4\\ 3 & -10\end{pmatrix}$ 为严格对角占优阵，故对此方程组用相应的雅可比迭代法和赛德尔迭代法皆收敛.

六、逐次超松弛迭代法

应用迭代法解方程组关键在于解决好收敛性问题，但即使迭代是收敛的，如果收敛速度太慢，这样的算法不能说是有效的，因此有必要发展新的方法，超松弛迭代法就是针对这一类问题发展起来的，在实用上发现，对于某些特殊矩阵，逐次超松弛迭代法的收敛速度比用雅可比迭代法和赛德尔迭代法快得多，是解大型稀疏矩阵方程组的有效方法之一.

解方程组 $\boldsymbol{AX}=\boldsymbol{b}$（设 $a_{ii}\neq 0, i=1,2,\cdots,n$）的逐次超松弛迭代法的迭代公式为

$$\begin{cases}x_i^{(m+1)}=x_i^{(m)}+\dfrac{\omega}{a_{ii}}\left(b_i-\sum\limits_{j=1}^{i-1}a_{ij}\cdot x_j^{(m+1)}-\sum\limits_{j=i}^{n}a_{ij}\cdot x_j^{(m)}\right)\\ \qquad (i=1,2,\cdots,n;\quad m=0,1,2,\cdots)\\ x^{(m)}=(x_1^{(m)},x_2^{(m)},\cdots,x_n^{(m)})^{\mathrm{T}}\end{cases}\tag{5.61}$$

其中，ω 为加速参数，称为松弛因子.

显然，当 $\omega=1$ 时，解 $\boldsymbol{AX}=\boldsymbol{b}$ 的 SOR 方法就是 G-S 迭代法，当 $0<\omega<1$ 时，迭代过程式(5.61)称为低松弛方法，当某些方程组用赛德尔迭代法不收敛时，可以用低松弛方法获得收敛，当 $\omega>1$ 时，迭代过程式(5.61)称为超松弛方法，可以用来加速收敛速度，并简称为 SOR 方法.

在 SOR 迭代法中，每迭代一次主要的运算量是计算一次矩阵与向量的乘法，且由式(5.61)可知用计算机计算时，应用 SOR 方法解 $\boldsymbol{AX}=\boldsymbol{b}$ 只需要一组工作单元，以便存放近似解 $\boldsymbol{X}^{(m)}$.

例 5.11 用逐次超松弛迭代法解方程组

$$\begin{cases} 4x_1+3x_2 \qquad\quad =24 \\ 3x_1+4x_2-\ x_3=30 \\ \qquad -\ x_2+4x_3=-24 \end{cases}$$

(方程组的精确解为 $\boldsymbol{X}=(3,4,-5)^{\mathrm{T}}$).

解 迭代公式为

$$\begin{cases} x_1^{(m+1)}=x_1^{(m)}+\omega(24-4x_1^{(m)}-3x_2^{(m)})/4 \\ x_2^{(m+1)}=x_2^{(m)}+\omega(30-3x_1^{(m+1)}-4x_2^{(m)}+x_3^{(m)})/4 \\ x_3^{(m+1)}=x_3^{(m)}+\omega(-24+x_2^{(m+1)}-4x_3^{(m)})/4 \end{cases}$$

$$(m=0,1,2,\cdots)$$

若取

$$\boldsymbol{X}^{(0)}=(1,1,1)^{\mathrm{T}}$$

则当取 $\omega=1$ 时(即 G-S 法)，解得

$$\begin{cases} \boldsymbol{X}^{(1)}=(5.250000,3.812500,-5.046875)^{\mathrm{T}} \\ \qquad\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \boldsymbol{X}^{(7)}=(3.0134110,3.9888241,-5.0027940)^{\mathrm{T}} \end{cases}$$

当取 $\omega=1.25$ 时

$$\begin{cases} \boldsymbol{X}^{(1)}=(6.312\,500,3.519\,531\,3,-6.650\,146\,5)^{\mathrm{T}} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \boldsymbol{X}^{(7)}=(3.000\,049\,8,4.000\,258\,6,-5.000\,348\,6)^{\mathrm{T}} \end{cases}$$

若要求迭代过程满足精度

$$\| \boldsymbol{X}^{(m)}-\boldsymbol{X}^{*} \|_{\infty}<\frac{1}{2}\times 10^{-7}$$

则用 G-S 方法(即 $\omega=1$)需要迭代 34 次,而用 SOR 方法($\omega=1.25$),仅需要迭代 14 次.

从这个简单的例子就可看到,松弛因子 ω 选得好,会使 SOR 方法的收敛大大加速.

下面推导 SOR 迭代法的迭代矩阵.

SOR 的迭代公式(5.61)可改写为

$$x_i^{(m+1)}=(1-\omega)x_i^{(m)}+\frac{\omega}{a_{ii}}\left(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(m+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(m)}\right)$$

$$(i=1,2,\cdots,n) \tag{5.62}$$

现在仍然将方程组的系数矩阵 $\boldsymbol{A}$ 拆为:

$$\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$$

其中,$\boldsymbol{D},\boldsymbol{L},\boldsymbol{U}$ 的意义同前.则式(5.62)可用矩阵表示为

$$\boldsymbol{X}^{(m+1)}=(1-\omega)\boldsymbol{X}^{(m)}+\omega\boldsymbol{D}^{-1}(\boldsymbol{b}+\boldsymbol{L}\boldsymbol{X}^{(m+1)}+\boldsymbol{U}\boldsymbol{X}^{(m)})$$

两边乘 $\boldsymbol{D}$ 得

$$\boldsymbol{D}\boldsymbol{X}^{(m+1)}=\boldsymbol{D}(1-\omega)\boldsymbol{X}^{(m)}+\omega\boldsymbol{L}\boldsymbol{X}^{(m+1)}+\omega\boldsymbol{U}\boldsymbol{X}^{(m)}+\omega\boldsymbol{b}$$

则
$$(\boldsymbol{D}-\omega\boldsymbol{L})\boldsymbol{X}^{(m+1)}=(1-\omega)\boldsymbol{D}\boldsymbol{X}^{(m)}+\omega\boldsymbol{U}\boldsymbol{X}^{(m)}+\omega\boldsymbol{b}$$

所以

$$\boldsymbol{X}^{(m+1)}=(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\{(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}\}\boldsymbol{X}^{(m)}$$

$$+(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\omega\boldsymbol{b}. \tag{5.63}$$

这就是 SOR 迭代公式(5.61)式的矩阵形式.

对照一般迭代法的迭代格式

$$\boldsymbol{X}^{(m+1)}=\boldsymbol{B}\boldsymbol{X}^{(m)}+\boldsymbol{g}$$

SOR 方法的迭代公式也可写为一般迭代公式形式

$$\boldsymbol{X}^{(m+1)}=\boldsymbol{B}_{\omega}\boldsymbol{X}^{(m)}+\boldsymbol{g}_{\omega}$$

其中 $$\boldsymbol{B}_{\omega}=(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\{(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}\}$$

称为 SOR 方法的迭代矩阵,所以说 SOR 方法也属于一般迭代法,因而前面关于一般迭代法的几个收敛定理,对 SOR 方法当然也适用,例如,由定理 5.9 知,逐次超松弛迭代法收敛的充要条件是

$$\rho(\boldsymbol{B}_{\omega})<1$$

但必须指出,改变松弛因子 ω 的值,不仅会影响迭代过程的收敛速度,而且还会影响迭代过程的敛散性.下面给出一些重要结论.

定理 5.12 逐次超松弛迭代法收敛的必要条件是松弛因子满足

$$0<\omega<2 \tag{5.64}$$

证明 因松弛法收敛,则由定理 5.9 知必有 $\rho(\boldsymbol{B}_{\omega})<1$,由矩阵 $\boldsymbol{B}_{\omega}$ 的特征方程的性质知

$$|\det(\boldsymbol{B}_{\omega})|=|\lambda_1\cdot\lambda_2\cdot\cdots\lambda_n|\leqslant[\rho(\boldsymbol{B}_{\omega})]^n$$

其中 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 是 $\boldsymbol{B}_{\omega}$ 的 n 个特征值

又 $$\det(\boldsymbol{B}_{\omega})=\det(\boldsymbol{D}-\boldsymbol{B}_{\omega})^{-1}\cdot\det[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}]$$

$$=\frac{1}{a_{11}a_{22}\cdots a_{nn}}(1-\omega)^n\cdot a_{11}a_{22}\cdots a_{nn}=(1-\omega)^n,$$

从而 $$|1-\omega|^n=\prod_{i=1}^{n}|\lambda_i|\leqslant[\rho(\boldsymbol{B}_{\omega})^n]<1$$

所以 $$|1-\omega|<1$$

即
$$0<\omega<2$$

上述定理说明,对于任何系数矩阵,若要松弛法收敛,必须选取松弛因子 $\omega\in(0,2)$. 然而,当松弛因子 ω 满足条件 $0<\omega<2$ 时,并不是对所有系数矩阵 $\boldsymbol{A}$ 松弛法均收敛,下面仍从 $\boldsymbol{A}$ 的特点给出收敛的充分判别条件.(不证明)

设有线性代数方程组

$$\boldsymbol{AX}=\boldsymbol{b}$$

[判别条件Ⅳ] 若 $\boldsymbol{A}$ 为对称正定阵,则当 $0<\omega<2$ 时,逐次超松弛迭代法收敛.

[判别条件Ⅴ] 若 $\boldsymbol{A}$ 为严格对角占优矩阵,则当 $0<\omega\leqslant1$ 时,逐次超松弛迭代法收敛.

如何选取 ω 使收敛速度最快?这自然是逐次超松弛法讨论的重要内容. 目前,只有少数特殊类型的矩阵,才有确定的最佳松弛因子的理论公式,但实际使用时也有一定困难. 通常的办法,是选不同的 ω 进行试算,以确定最佳 ω 的近似值;或者先取一个 $\omega(0<\omega<2)$,然后根据迭代过程收敛的快慢不断修改 ω,这样逐步寻找最佳的 ω,直到满意后再固定下来继续迭代,以达到加速的目的.

对线性代数方程组使用迭代法求解还需注意以下几点:

① 虽然迭代法的收敛性与初始向量的选取无关,但初始向量的选取对迭代的工作量却有重大影响. 可以证明,当初始向量 $\boldsymbol{X}^{(0)}$ 取得接近于精确解 $\boldsymbol{X}$ 时,只要较少次数的迭代就能达到精度要求,因而应重视初始向量的选取.

② 在迭代过程中,舍入误差的影响仍然存在,尽管 $\boldsymbol{X}^{(k)}\to\boldsymbol{X}^*$,但由于机器字长的限制,不可能达到任意的精度,最多只能达到机器精度,因而在使用实用误差估计式 $\|\boldsymbol{X}^{(m)}-\boldsymbol{X}^{(m-1)}\|<\varepsilon$ 来停止迭代时,ε 要选得恰当,小于或近于机器的精度,都可能造成死循环.

③ 在迭代格式 $\boldsymbol{X}^{(m+1)}=\boldsymbol{BX}^{(m)}+\boldsymbol{g}$ 中,若 $\boldsymbol{I}-\boldsymbol{B}$ 是病态矩阵,那

么一般迭代法也得不到好的结果. 例如:

$$\boldsymbol{B}=\begin{bmatrix} 0 & 0.999\,999 \\ 0.999\,999 & 0 \end{bmatrix}, \quad \boldsymbol{g}=\begin{bmatrix} 10^{-6} \\ 10^{-6} \end{bmatrix}$$

其精确解为 $\boldsymbol{X}=(1,1)^{\mathrm{T}}$

若取初始向量

$$\boldsymbol{X}^{(0)}=\begin{bmatrix} 0.600\,000 \\ 0.600\,000 \end{bmatrix}$$

进行迭代,则有

$$\boldsymbol{X}^{(0)}=\boldsymbol{X}^{(1)}=\boldsymbol{X}^{(2)}=\cdots$$

所以方程组的近似解 $\tilde{\boldsymbol{X}}=\boldsymbol{X}^{(0)}$,但它与方程组的精确解 $\boldsymbol{X}$ 相差很大,得到如此不好的迭代结果的原因就是因为 $\boldsymbol{I}-\boldsymbol{B}$ 是病态矩阵所引起的,事实上,容易计算 $\boldsymbol{I}-\boldsymbol{B}$ 的条件数

$$\mathrm{Cond}(\boldsymbol{I}-\boldsymbol{B})_{\infty}=2\times 10^{6}$$

相当大,所以迭代效果不好.

学习指导

一、基本要求与重点

1. 掌握解线性代数方程组的直接法的几种基本常用数值方法,例如,按列选主元的高斯消去法、三角分解法等,并能比较它们各自的优缺点,能熟练地对方阵 A 进行杜利脱尔(Doolittle)分解、克洛脱(Crout)分解和乔列斯基(Cholesky)分解.

2. 掌握用迭代法求解线性代数方程组的基本思想和计算步骤,能熟练地写出雅可比(Jacobi)迭代法(简称 J-迭代法),高斯-赛德尔(Gauss-Seidel)迭代法(简称 G-S 迭代法)和逐次超松弛迭

代法(简称 SOR 迭代法)的迭代格式的分量形式和矩阵形式，并能比较它们各自的特点.

3. 了解向量范数和矩阵范数的概念，并会计算几种常用的向量和矩阵的范数.

4. 熟练掌握常用的判别 J-迭代法、G-S 迭代法和 SOR 迭代法收敛性的各种判别条件，并学会判断给定迭代式的收敛性及误差估计.

5. 了解方程的状态和系数矩阵的条件数的关系，会计算方阵 $\boldsymbol{A}$ 的条件数，并对方程组进行初步“扰动分析”.

二、例题分析与解答

例 1 采用四位有效数字，分别用顺序高斯消去法和列主元素高斯消去法求解线性代数方程组：

$$\begin{bmatrix} 0.7290 & 0.8100 & 0.9000 \\ 1.0000 & 1.0000 & 1.0000 \\ 1.3310 & 1.2100 & 1.1000 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.6867 \\ 0.8338 \\ 1.0000 \end{bmatrix}$$

解 用代数方法可求得此方程组的精确解为

$$x_1=0.2245,\ x_2=0.2814,\ x_3=0.3279$$

(1) 用顺序高斯消去法求解

消元过程用矩阵可表示为

$$\left[\begin{array}{ccc:c} 0.7290 & 0.8100 & 0.9000 & 0.6867 \\ 1.000 & 1.000 & 1.000 & 0.8338 \\ 1.331 & 1.210 & 1.100 & 1.00 \end{array}\right] \xrightarrow[l_{21}=1.372,\ l_{31}=1.826]{\text{第一次消元}}$$

$$\left[\begin{array}{ccc:c} 0.7290 & 0.8100 & 0.9000 & 0.6867 \\ & -0.1110 & -0.2350 & -0.1084 \\ & -0.2690 & -0.5430 & -0.2540 \end{array}\right] \xrightarrow[]{\substack{\text{第二次消元} \\ l_{32}=2.4230}}$$

$$
\left[\begin{array}{ccc:c}
0.7290 & 0.8100 & 0.9000 & 0.6867 \\
 & -0.1110 & -0.2350 & -0.1084 \\
 & & 0.02640 & 0.008700
\end{array}\right]
$$

回代求解得：$x_1 = 0.2252$，$x_2 = 0.2790$，$x_3 = 0.3295$

它与精确解比较，显然误差很大.

（2）用列主元素高斯消去法求解

下面增广矩阵中打框 $\boxed{}$ 的元素，是每步消元的主元素.

$$
\left[\begin{array}{ccc:c}
0.7290 & 0.8100 & 0.9000 & 0.6867 \\
1.000 & 1.000 & 1.000 & 0.8338 \\
\boxed{1.3131} & 1.210 & 1.100 & 1.000
\end{array}\right]
\xrightarrow[\substack{l_{21}=0.7513 \\ l_{31}=0.5477}]{r_1 \leftrightarrow r_3}
$$

$$
\left[\begin{array}{ccc:c}
1.331 & 1.210 & 1.100 & 1.000 \\
 & 0.09090 & 0.1736 & 0.08250 \\
 & \boxed{0.1473} & 0.2975 & 0.1390
\end{array}\right]
\xrightarrow[l_{32}=0.6171]{r_2 \leftrightarrow r_3}
$$

$$
\left[\begin{array}{ccc}
1.331 & 1.210 & 1.100 \\
 & 0.1473 & 0.2975 \\
 & & -0.01000
\end{array}\right]
$$

回代求解得：$x_1 = 0.2246$，$x_2 = 0.2812$，$x_3 = 0.3280$

它与方程组的精确解已相当接近了，是较好的近似解. 与不选主元的顺序高斯法相比，显然用列主元素高斯消去法可以得到较好的数值解.

列主元素消去法方法简单，精度高，是机器上解中、小型线代数方程组常用的方法.

例 2 用直接三角分解法求解方程组

$$
\begin{cases}
2x_1+4x_2+2x_3+6x_4=9\\
4x_1+9x_2+6x_3+15x_4=23\\
2x_1+6x_2+9x_3+18x_4=22\\
6x_1+15x_2+18x_3+40x_4=47
\end{cases}
$$

解　方程组的系数矩阵

$$
\boldsymbol{A}=\begin{bmatrix}2&4&2&6\\4&9&6&15\\2&6&9&18\\6&15&18&40\end{bmatrix}\text{，右端项 }\boldsymbol{b}=\begin{bmatrix}9\\23\\22\\47\end{bmatrix}
$$

（1）**分解 $\boldsymbol{A}=\boldsymbol{LU}$**

由公式(5.12)**得**

$$
\boldsymbol{A}=\begin{bmatrix}1&&&\\2&1&&\\1&2&1&\\3&3&2&1\end{bmatrix}\begin{bmatrix}2&4&2&6\\&1&2&3\\&&3&6\\&&&1\end{bmatrix}
$$

（2）**求解 $\boldsymbol{LY}=\boldsymbol{b}$，即**

$$
\begin{bmatrix}1&&&\\2&1&&\\1&2&1&\\3&3&2&1\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}=\begin{bmatrix}9\\23\\22\\47\end{bmatrix}
$$

由公式(5.13)**得到**　$\boldsymbol{Y}=(9,5,3,-1)^{\mathrm{T}}$.

（3）**求解 $\boldsymbol{UX}=\boldsymbol{Y}$，即**

$$
\begin{bmatrix}2&4&2&6\\&1&2&3\\&&3&6\\&&&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}9\\5\\3\\-1\end{bmatrix}
$$

由公式(5.14)可解得　$\boldsymbol{X}=(0.5,2,3,-1)^{\mathrm{T}}$

即 $x_1=0.5, x_2=2, x_3=3, x_4=-1$.

例 3　用直接三角分解法求解两个有相同系数矩阵的方程组 $\boldsymbol{AX}=\boldsymbol{b}_1$ 和 $\boldsymbol{AX}=\boldsymbol{b}_2$.

其中　$\boldsymbol{A}=\begin{bmatrix}1 & 2 & 3 & 4\\ 1 & 4 & 9 & 16\\ 1 & 8 & 27 & 64\\ 1 & 16 & 81 & 256\end{bmatrix}$, $\boldsymbol{b}_1=\begin{bmatrix}4\\ 10\\ 28\\ 82\end{bmatrix}$, $\boldsymbol{b}_2=\begin{bmatrix}2\\ 12\\ 56\\ 240\end{bmatrix}$

解　用系数矩阵 $\boldsymbol{A}$ 的 $\boldsymbol{LU}$ 直接三角分解法求解方程组时，由于把对系数矩阵的计算和对右端项的计算分开了，这就使我们在计算系数矩阵相同而右端项不同的若干方程组：

$$\boldsymbol{AX}=(\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m)$$

时更显得方便(其中，$\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m$ 是各方程组的右端向量)，只须做一次矩阵分解 $\boldsymbol{A}=\boldsymbol{LU}$，然后解 m 个三角形方程组：

$$\boldsymbol{UX}=\boldsymbol{Y}_i, \ (i=1,2,\cdots,m)$$

即可. 本题中可先由 $\boldsymbol{A}, \boldsymbol{b}_1, \boldsymbol{b}_2$ 组成 4×6 阶矩阵：

$$[\boldsymbol{A}, \boldsymbol{b}_1, \boldsymbol{b}_2]=\left[\begin{array}{cccc:cc}1 & 2 & 3 & 4 & 4 & 2\\ 1 & 4 & 9 & 16 & 10 & 12\\ 1 & 8 & 27 & 64 & 28 & 56\\ 1 & 16 & 81 & 256 & 82 & 240\end{array}\right]$$

$$\xrightarrow[\text{分解得}]{\text{由公式(5.12)}}\left[\begin{array}{cccc:cc}1 & 2 & 3 & 4 & 4 & 2\\ 1 & 2 & 6 & 12 & 6 & 10\\ 1 & 3 & 6 & 24 & 6 & 24\\ 1 & 7 & 6 & 24 & 0 & 24\end{array}\right]$$

即得

$$
U\begin{bmatrix}1 & 2 & 3 & 4\\ & 2 & 6 & 12\\ & & 6 & 24\\ & & & 24\end{bmatrix},\ Y_1=\begin{bmatrix}4\\6\\6\\0\end{bmatrix},\ Y_2=\begin{bmatrix}2\\10\\24\\24\end{bmatrix}
$$

再用回代分别求解 $UX=Y_1$, $UX=Y_2$,,可得题中两个方程组的解分别为

$$
X=(1,0,1,0)^{\mathrm{T}},\ X_2=(0,-1,0,1)^{\mathrm{T}}.
$$

例 4　用平方根法求解方程组

$$
\begin{bmatrix}1 & 1 & 1 & 1 & 1\\1 & 2 & 2 & 2 & 2\\1 & 2 & 3 & 3 & 3\\1 & 2 & 3 & 4 & 4\\1 & 2 & 3 & 4 & 5\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=\begin{bmatrix}5\\9\\12\\14\\15\end{bmatrix}
$$

解　系数矩阵 A 显然是对称正定的,可用平方根法求解.

(1) 对 A 进行乔列斯基分解:$A=LL^{\mathrm{T}}$

由公式(5.20),按列确定下三角阵 L,可得

$$
L=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\1 & 1 & 0 & 0 & 0\\1 & 1 & 1 & 0 & 0\\1 & 1 & 1 & 1 & 0\\1 & 1 & 1 & 1 & 1\end{bmatrix}
$$

(2) 求解下三角形方程组 $LY=b$,即

$$
\begin{bmatrix}1 & 0 & 0 & 0 & 0\\1 & 1 & 0 & 0 & 0\\1 & 1 & 1 & 0 & 0\\1 & 1 & 1 & 1 & 0\\1 & 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\\y_5\end{bmatrix}=\begin{bmatrix}5\\9\\12\\14\\15\end{bmatrix}
$$

由公式(5.21)自上而下可求得 $Y=(5,4,3,2,1)^{T}$.

(3) 求解上三角形方程组 $L^{T}X=Y$

由公式(5.22)自下而上回代可求得 $X=(1,1,1,1,1)^{T}$.

例 5 用系数矩阵 A 的克洛脱分解求解方程组

$$\begin{cases}6x_1+2x_2+x_3-x_4=6\\2x_1+4x_2+x_3=-1\\x_1+x_2+4x_3-x_4=5\\-x_1-x_3+3x_4=-5\end{cases}$$

解 方程组的系数矩阵 A 和右端向量 b 分别为

$$A=\begin{bmatrix}6&2&1&-1\\2&4&1&0\\1&1&4&-1\\-1&0&-1&3\end{bmatrix},\ b=\begin{bmatrix}6\\-1\\5\\-5\end{bmatrix}$$

(1) 对 A 进行克洛脱分解,得 $A=\widetilde{L}\widetilde{U}$,式中,$\widetilde{L}$ 是下三角阵,$\widetilde{U}$ 是单位上三角阵,即

$$A=\begin{bmatrix}l_{11}&&&\\l_{21}&l_{22}&&\\l_{31}&l_{32}&l_{33}&\\l_{41}&l_{42}&l_{43}&l_{44}\end{bmatrix}\begin{bmatrix}1&u_{12}&u_{13}&u_{14}\\&1&u_{23}&u_{24}\\&&1&u_{34}\\&&&1\end{bmatrix}$$

利用矩阵乘法比较两边元素,得

$$\widetilde{L}=\begin{bmatrix}6&&&\\2&\frac{10}{3}&&\\1&\frac{2}{3}&\frac{37}{10}&\\-1&\frac{1}{3}&-\frac{9}{10}&\frac{191}{74}\end{bmatrix},\ \widetilde{U}=\begin{bmatrix}1&\frac{1}{3}&\frac{1}{6}&-\frac{1}{6}\\0&1&\frac{1}{5}&\frac{1}{10}\\0&0&1&-\frac{9}{37}\\0&0&0&1\end{bmatrix}$$

（2）求解下三角形方程组 $\widetilde{L}Y=b$

自上而下可逐个求得 Y 的全部分量，得 $Y=\left(1,\ -\frac{9}{10},\ \frac{46}{37},\ -1\right)^{\mathrm{T}}$

（3）求解上三角形方程组 $\widetilde{U}X=Y$

自下而上回代可得　$X=(1,-1,1,-1)^{\mathrm{T}}$

即方程组的解为　$x_1=1, x_2=-1, x_3=1, x_4=-1.$

例 6　用追赶法求解三对角线性代数方程组 $AX=d$，其中

$$A=\begin{bmatrix} 2 & 2 & 0 & 0 & 0 \\ -1 & 1 & 2 & 0 & 0 \\ 0 & -1 & 1 & 2 & 0 \\ 0 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix},\ d=\begin{bmatrix} 6 \\ 7 \\ 9 \\ 11 \\ 1 \end{bmatrix}$$

解　此方程组的系数矩阵 A 是三对角阵.

（1）对 A 进行克洛脱分解：$A=\widetilde{L}\widetilde{U}$

由公式(5.29)得

$$\widetilde{L}=\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix},\ \widetilde{U}=\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

（2）求解 $\widetilde{L}Y=d$

由公式(5.30)得　$Y=(3,5,7,9,5)^{\mathrm{T}}$

（3）求解 $\widetilde{U}X=Y$

由公式(5.31)得　$X=(1,2,3,4,5)^{\mathrm{T}}$，即为方程组的解.

例 7　举例说明一个非奇异矩阵不一定存在 LU 分解.

解　取 $A=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$，显然 A 是非奇异矩阵，若 A 存在 LU 分解

(L 为单位下三角阵,U 为上三角阵),则有

$$\boldsymbol{A}=\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}=\begin{bmatrix}1 & 0\\a & 1\end{bmatrix}\begin{bmatrix}b & c\\0 & d\end{bmatrix}=\begin{bmatrix}b & c\\ab & ac+d\end{bmatrix}$$

由矩阵相等则必对应元素相等,从而得 $b=0$,则 $ab=0$,这与 $ab=1$ 矛盾,所以 $\boldsymbol{A}=\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$尽管是非奇异矩阵,但不存在 $\boldsymbol{A}=\boldsymbol{LU}$ 分解.

例 8 在二维向量空间 $\boldsymbol{R}^2$ 中,图示下列三种常用向量范数的点集:

$$\|\boldsymbol{X}\|_1\leqslant 1,\quad \|\boldsymbol{X}\|_2\leqslant 1,\quad \|\boldsymbol{X}\|_\infty\leqslant 1.$$

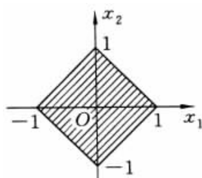
解 设 $\boldsymbol{X}=(x_1,x_2)^{\mathrm{T}}\in\boldsymbol{R}_2$ 则有

$$\|\boldsymbol{X}\|_1=|x_1|+|x_2|\leqslant 1,$$

$$\|\boldsymbol{X}\|_2=\sqrt{x_1^2+x_2^2}\leqslant 1,$$

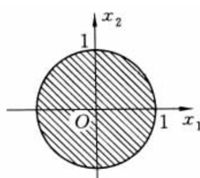
$$\|\boldsymbol{X}\|_\infty=\max\{|x_1|,|x_2|\}\leqslant 1$$

它们在平面上的图形分别如图(例)5-1、图(例)5-2 和图(例)5-3 所示.



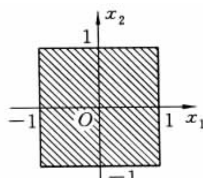
$\|\boldsymbol{X}\|_1\leqslant 1$(菱形)

图(例)5-1



$\|\boldsymbol{X}\|_2\leqslant 1$(单位圆)

图(例)5-2



$\|\boldsymbol{X}\|_\infty\leqslant 1$(正方形)

图(例)5-3

这些点集的共同点是:它们都是有界的、闭的,并且关于原点对称的内部非空的集合.

例 9 计算方阵

$$A=\begin{bmatrix}1 & 0 & 0\\0 & 2 & 4\\0 & -2 & 4\end{bmatrix}$$**的三种常用范数** $\|A\|_P (P=1,2,\infty)$

解 $\|A\|_1=\max\limits_{1\leqslant j\leqslant 3}\sum\limits_{i=1}^{3}|a_{ij}|=\max\{1,4,8\}=8$

$$\|A\|_\infty=\max_{1\leqslant i\leqslant 3}\sum_{j=1}^{3}|a_{ij}=\max\{1,6,6\}=6$$

由 $\|A\|_2=\sqrt{\lambda_{\max}(A^TA)}$**,先计算**

$$A^TA=\begin{bmatrix}1 & 0 & 0\\0 & 2 & -2\\0 & 4 & 4\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\0 & 2 & 4\\0 & -2 & 4\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\0 & 8 & 0\\0 & 0 & 32\end{bmatrix}$$

所以 $\lambda_{\max}(A^TA)=32$,

从而 $\|A\|_2=\sqrt{32}=4\sqrt{2}$.

例 10 求解线性代数方程组:$\begin{cases}x_1+10^4x_2=10^4\\x_1+x_2=2\end{cases}$

解 显然,本题的精确解为 $x_1=1,x_2=1$,**系数矩阵为** $A=\begin{bmatrix}1 & 10^4\\1 & 1\end{bmatrix}$,**在一般情况下,条件数** Cond($A$)**很大的方阵** A **为病态阵,以病态阵为系数矩阵的方程组,称为病态方程组,在病态方程组求解时,解对于初始数据是非常敏感的. 本题中,因**

$$A^{-1}=\frac{1}{10^4-1}\begin{bmatrix}-1 & 10^4\\1 & -1\end{bmatrix}$$

从而条件数 $\mathrm{Cond}_\infty(A)=\dfrac{(10^4+1)^2}{10^4-1}\approx 10^4$ **非常大,所以方程组是**

病态方程组，即使用高斯按列选主元消去法计算，最终求得的解是 $x_1=0, x_2=1$，严重失真.

对于病态方程组的解法，应用上，一般可用高精度的算术运算，以减轻病态矩阵对解的影响. 有时，若选取一个适当的数去乘方程组中的某个方程，以达到改变系数矩阵的条件数，则其解的精度可能会有所改善.

本题中若用 10^{-4} 乘第一个方程，则方程组的系数矩阵变为

$$\boldsymbol{B}=\begin{bmatrix}10^{-4} & 1\\ 1 & 1\end{bmatrix},\ \boldsymbol{B}^{-1}\frac{1}{1-10^{-4}}\begin{bmatrix}-1 & 1\\ 1 & -10^{-4}\end{bmatrix}$$

从而条件数 $\mathrm{Cond}_{\infty}(\boldsymbol{B})=\dfrac{4}{1-10^{-4}}\approx 4$，比原方程组的条件数大大减少，若用按列选主元的高斯消去法求解，可得 $x_1=1, x_2=1$，与精确解比较，显然是一个很好的数值解.

例 11 分别用雅可比迭代法与高斯-赛德尔迭代法求解方程组

$$\begin{cases}10x_1+3x_2+x_3=14\\ 2x_1-10x_2+3x_3=-5\\ x_1+3x_2+10x_3=14\end{cases}$$

取初值 $\boldsymbol{X}^{(0)}=(0,0,0)^{\mathrm{T}}$，精确到小数后四位.

解 显然，本题的精确解为 $x_1=1, x_2=1, x_3=1$.

(1) 用雅可比方法计算，由公式(5.47)得其迭代公式为

$$\begin{cases}x_1^{(m+1)}=(-3x_2^{(m)}-x_2^{(m)}+14)/10\\ x_2^{(m+1)}=(-2x_1^{(m)}-3x_3^{(m)}-5)/(-10)\\ x_3^{(m+1)}=(-x_1^{(m)}-3x_2^{(m)}+14)/10,\ (m=0,1,2,\cdots)\end{cases}$$

当取 $\boldsymbol{X}^{(0)}=(0,0,0)^{\mathrm{T}}$ 迭代六次的计算值为

表(例)5-1

m	1	2
$\boldsymbol{X}^{(m)}$	$(1.4,0.5,1.4)^{T}$	$(1.11,1.20,1.11)^{T}$
m	3	4
$\boldsymbol{X}^{(m)}$	$(0.929,1.055,0.929)^{T}$	$(0.9906,0.9645,0.9906)^{T}$
m	5	6
$\boldsymbol{X}^{(m)}$	$(1.0116,0.9953,1.0116)^{T}$	$(1.000251,1.005795,1.000251)^{T}$

(2) 用高斯-赛德尔迭代法计算,由公式(5.48)得其迭代公式为

$$\begin{cases} x_1^{(m+1)}=(-3x_2^{(m)}-x_3^{(m)}+14)/10 \\ x_2^{(m+1)}=(-2x_1^{(m+1)}-3x_3^{(m)}-5)/(-10) \\ x_3^{(m+1)}=(-x_1^{(m+1)}-3x_2^{(m+1)}+14)/10, \ (m=0,1,2,\cdots) \end{cases}$$

同样取初值 $\boldsymbol{X}^{(0)}=(0,0,0)^{T}$,迭代四次就可得 $\boldsymbol{X}^{(4)}=(0.99154, 0.99578, 1.0021)^{T}$,从而可见,本题用高斯-赛德尔迭代法显然比用雅可比迭代法收敛快.

例 12 分别用雅可比迭代法、高斯-赛德尔迭代法及逐次超松弛迭代法计算线性代数方程组

$$\begin{cases} -4x_1+x_2+x_3+x_4=1 \\ x_1-4x_2+x_3+x_4=1 \\ x_1+x_2-4x_3+x_4=1 \\ x_1+x_2+x_3-4x_4=1 \end{cases}$$

取初值 $\boldsymbol{X}^{(0)}=(0,0,0,0)^{\mathrm{T}}$，精确到 10^{-5}.

解 本题的精确解为 $x_1=x_2=x_3=x_4=-1$，

(1) 雅可比迭代公式的分量形式为

$$\begin{cases} x_1^{(m+1)}=-\dfrac{1}{4}(1-x_2^{(m)}-x_3^{(m)}-x_4^{(m)}) \\ x_2^{(m+1)}=-\dfrac{1}{4}(1-x_1^{(m)}-x_3^{(m)}-x_4^{(m)}) \\ x_3^{(m+1)}=-\dfrac{1}{4}(1-x_1^{(m)}-x_2^{(m)}-x_4^{(m)}) \\ x_4^{(m+1)}=-\dfrac{1}{4}(1-x_1^{(m)}-x_2^{(m)}-x_3^{(m)}) \end{cases}$$

$$(m=0,1,2,\cdots)$$

则迭代 43 次才得数值解为

$$\boldsymbol{X}^{(43)}=(-1.00000,-1.00000,-1.00000,-1.00000)^{\mathrm{T}}.$$

(2) 高斯-赛德尔迭代公式的分量形式为

$$\begin{cases} x_1^{(m+1)}=-\dfrac{1}{4}(1-x_2^{(m)}-x_3^{(m)}-x_4^{(m)}) \\ x_2^{(m+1)}=-\dfrac{1}{4}(1-x_1^{(m+1)}-x_3^{(m)}-x_4^{(m)}) \\ x_3^{(m+1)}=-\dfrac{1}{4}(1-x_1^{(m+1)}-x_2^{(m+1)}-x_4^{(m)}) \\ x_4^{(m+1)}=-\dfrac{1}{4}(1-x_1^{(m+1)}-x_2^{(m+1)}-x_3^{(m+1)}) \end{cases}$$

$$(m=0,1,2,\cdots)$$

则迭代到第 22 次就可得数值解为

$$\boldsymbol{X}^{(22)}=(-0.99999,-0.99999,-1.00000,-1.00000)^{\mathrm{T}}.$$

(3) 逐次超松弛迭代公式的分量形式为

$$\begin{cases} x_1^{(m+1)} = x_1^{(m)} - \dfrac{\omega}{4}(1 + 4x_1^{(m)} - x_2^{(m)} - x_3^{(m)} - x_4^{(m)}) \\ x_2^{(m+1)} = x_2^{(m)} - \dfrac{\omega}{4}(1 - x_1^{(m+1)} + 4x_2^{(m)} - x_3^{(m)} - x_4^{(m)}) \\ x_3^{(m+1)} = x_3^{(m)} - \dfrac{\omega}{4}(1 - x_1^{(m+1)} - x_2^{(m+1)} + 4x_3^{(m+1)} - x_4^{(m)}) \\ x_4^{(m+1)} = x_4^{(m)} - \dfrac{\omega}{4}(1 - x_1^{(m+1)} - x_2^{(m+1)} - x_3^{(m+1)} + 4x_4^{(m)}) \end{cases}$$

$$(m=0,1,2,\cdots)$$

当 ω 取不同值时，加速的效果是不一样的，下面分别列举一些不同 ω 取值的计算结果.

例如取 $\omega=1.0$ 时，迭代 21 次，得 $\boldsymbol{X}^{(21)}=(-0.99999, -0.99999, -1.00000, -1.00000)^{\mathrm{T}}$；

取 $\omega=1.1$ 时，迭代 16 次，得 $\boldsymbol{X}^{(16)}=(-0.99999, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.2$ 时，迭代 11 次，得 $\boldsymbol{X}^{(11)}=(-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.3$ 时，迭代 10 次，得 $\boldsymbol{X}^{(10)}=(-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.4$ 时，迭代 13 次，得 $\boldsymbol{X}^{(13)}=(-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.5$ 时，迭代 16 次，得 $\boldsymbol{X}^{(16)}=(-0.99999, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.6$ 时，迭代 22 次，得 $\boldsymbol{X}^{(22)}=(-1.00000, -1.00000, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.7$ 时，迭代 32 次，得 $\boldsymbol{X}^{(32)}=(-1.00000, -0.99999, -1.00000, -1.00000)^{\mathrm{T}}$；

$\omega=1.8$ 时，迭代 52 次，得 $\boldsymbol{X}^{(52)}=(-0.99999, -1.00000, -1.00000, -0.99999)^{\mathrm{T}}$.

显然，松弛因子 ω 选得好，会使逐次超松弛迭代法的收敛速度大大加快，本题选 $\omega=1.3$ 时，加速效果最明显.

例 13 设线性代数方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的系数矩阵为

$$\boldsymbol{A}=\begin{bmatrix}4 & 3 & 0\\ 3 & 4 & -1\\ 0 & -1 & 4\end{bmatrix}$$

若用逐次超松弛迭代法求解，试确定其最佳松弛因子 ω.

解 逐次超松弛迭代法的迭代矩阵 $\boldsymbol{B}_\omega$ 与松弛因子 ω 有关，当取不同的 ω 时，则其迭代收敛速度就不同，因此，如何选取最佳的松弛因子 ω 使迭代收敛的速度最快，这自然是逐次超松弛法讨论的主要内容. 应用中通常的办法是选不同的 ω 进行试算，以确定最佳 ω 的近似值；或者先取一个 $\omega(0<\omega<2)$，然后根据迭代过程收敛的快慢，不断修改 ω，这样逐步寻找最佳 ω，直到满意后再固定下来继续迭代，以达到加速的目的. 但对于一类有特殊性质的矩阵（它们常在偏微分方程的数值解法中出现），已有确定的最佳松弛因子的理论公式. 例如

定理 如果 $\boldsymbol{A}$ 是对称正定阵，且还是三对角阵，则逐次超松弛方法的最佳松弛因子为

$$\omega=\frac{2}{1+\sqrt{1-[\rho(\boldsymbol{B}_\mathrm{J})]^2}}$$

其中，$\rho(\boldsymbol{B}_\mathrm{J})$ 是雅可比迭代方法的迭代矩阵 $\boldsymbol{B}_\mathrm{J}$ 的谱半径.

本题中的 $\boldsymbol{A}$ 是对称正定阵，且是非奇异三对角阵，应用上述定理，首先要计算 $\rho(\boldsymbol{B}_\mathrm{J})$，由雅可比迭代矩阵 $\boldsymbol{B}_\mathrm{J}$ 为

$$\boldsymbol{B}_\mathrm{J}=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})=\begin{bmatrix}\frac{1}{4} & & \\ & \frac{1}{4} & \\ & & \frac{1}{4}\end{bmatrix}\begin{bmatrix}0 & -3 & 0\\ -3 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}$$

$$=\begin{bmatrix} 0 & -0.75 & 0 \\ -0.75 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix}$$

于是由 $\det(\boldsymbol{B}_J-\lambda\boldsymbol{I})=-\lambda(\lambda^2-0.625)=0$，得 $\rho(\boldsymbol{B}_J)=\sqrt{0.625}$，故

$$\omega=\frac{2}{1+\sqrt{1-[\rho(\boldsymbol{B})_J]^2}}=\frac{2}{1+\sqrt{1-0.625}}\approx 1.24$$ 是所求的最佳松弛因子. 并且可算得 $\rho(\boldsymbol{B}_\omega)\approx 0.24$，与 $\rho(\boldsymbol{B}_J)$，$\rho(\boldsymbol{B}_G)$ 比较，显然采用最佳松弛因子的逐次超松弛方法要比高斯-赛德尔迭代法和雅可比迭代法收敛快得多.

例 14 分别用雅可比迭代法与高斯-赛德尔迭代法求解方程组

$$\begin{cases} 20x_1+2x_2+3x_3=1 \\ x_1+8x_2+x_3=2 \\ 2x_1-3x_2+15x_3=3 \end{cases}$$

取 $\boldsymbol{X}^{(0)}=(0,0,0)^T$，问迭代法是否收敛？若收敛，需要迭代多少次，才能保证各分量的误差绝对值小于 10^{-6}？

解 雅可比迭代法的迭代公式的分量形式为

$$\begin{cases} x_1^{(m+1)}=\dfrac{1}{20}(-2x_2^{(m)}-3x_3^{(m)}+24) \\ x_2^{(m+1)}=\dfrac{1}{8}(-x_1^{(m)}-x_3^{(m)}+12) \\ x_3^{(m+1)}=\dfrac{1}{15}(-2x_1^{(m)}+3x_2^{(m)}+30) \end{cases}$$

其迭代矩阵为

$$\boldsymbol{B}_J=\begin{bmatrix} 0 & -\dfrac{1}{10} & -\dfrac{3}{20} \\ -\dfrac{1}{8} & 0 & -\dfrac{1}{8} \\ -\dfrac{2}{15} & \dfrac{1}{5} & 0 \end{bmatrix}$$

由于 $\|\boldsymbol{B}_{\mathrm{J}}\|_{\infty}=\frac{1}{3}<1$,所以用雅可比迭代法一定收敛.

若取 $\boldsymbol{X}^{(0)}=(0,0,0)^{\mathrm{T}}$,经一次迭代得 $\boldsymbol{X}^{(1)}=\left(\frac{6}{5},\frac{3}{2},2\right)^{\mathrm{T}}$,则 $\|\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)}\|_{\infty}=2$. 由公式(5.23)可得

$$m>\left[\ln\frac{10^{-6}\left(1-\frac{1}{3}\right)}{2}\Big/\ln\frac{1}{3}\right]=13$$

也即,要保证各分量误差绝对值小于 10^{-6},需要迭代 14 次.

高斯-赛德尔迭代法的迭代公式的分量形式为

$$\begin{cases}x_1^{(m+1)}=\frac{1}{20}(-2x_2^{(m)}-3x_3^{(m)}+24)\\ x_2^{(m+1)}=\frac{1}{8}(-x_1^{(m+1)}-x_3^{(m)}+12)\\ x_3^{(m+1)}=\frac{1}{15}(-2x_1^{(m+1)}+3x_2^{(m+1)}+30)\end{cases}$$

其迭代矩阵为

$$\boldsymbol{B}_{\mathrm{G}}=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}=\frac{1}{2\,400}\begin{bmatrix}0 & -240 & -360\\ 0 & 30 & -25.5\\ 0 & 38 & -3\end{bmatrix}$$

由于 $\|\boldsymbol{B}_{\mathrm{G}}\|_{\infty}=\frac{1}{4}<1$,所以用高斯-赛德尔迭代法也一定收敛. 取 $\boldsymbol{X}^{(0)}=(0,0,0)^{\mathrm{T}}$,经一次迭代得 $\boldsymbol{X}^{(1)}=(1.2,1.35,2.11)^{\mathrm{T}}$,则 $\|\boldsymbol{X}^{(1)}-\boldsymbol{X}^{(0)}\|_{\infty}=2.11$. 由公式(5.56)可得

$$m>\left[\ln\frac{10^{-6}\left(1-\frac{1}{4}\right)}{2.11}\Big/\ln\frac{1}{4}\right]=10.$$

也即,要保证各分量误差绝对值小于 10^{-6},只要迭代 11 次.

例 15 设线性代数方程组 $\boldsymbol{A}\boldsymbol{X}=\boldsymbol{b}$ 的系数矩阵为

$$A=\begin{bmatrix}1 & a & a\\ a & 1 & a\\ a & a & 1\end{bmatrix}$$

证明：(1) **当** $-0.5<a<1$ **时，用高斯-赛德尔迭代法求解** $\boldsymbol{AX}=\boldsymbol{b}$ **收敛；**

(2) **当** $-0.5<a<0.5$ **时，用雅可比迭代法求解** $\boldsymbol{AX}=\boldsymbol{b}$ **收敛.**

证 当 $-0.5<a<1$ 时，$\boldsymbol{A}$ 的三个顺序主子式均大于零，即

$$\Delta_1=1>0,\Delta_2=\begin{vmatrix}1 & a\\ a & 1\end{vmatrix}=1-a^2>0,\Delta_3=\begin{vmatrix}1 & a & a\\ a & 1 & a\\ a & a & 1\end{vmatrix}=1+2a^2-3a^2>0.$$

所以，$\boldsymbol{A}$ 是对称正定的，由[判别条件Ⅱ]知，用高斯-赛德尔迭代法求解 $\boldsymbol{AX}=\boldsymbol{b}$ 是一定收敛的.

当 $-0.5<a<0.5$ 时，$\boldsymbol{A}$ 为按行严格对角占优. 由[判别条件Ⅰ]知，用雅可比迭代法求解 $\boldsymbol{AX}=\boldsymbol{b}$ 是一定收敛的，且若用高斯-赛德尔迭代法也一定收敛.

此题中，若取 $a=0.8$，则高斯-赛德尔迭代收敛，而雅可比迭代发散. 这是因为 $2\boldsymbol{D}-\boldsymbol{A}$ 不是正定的，由[判别条件Ⅲ]即得.

例 16 **设线性代数方程组** $\boldsymbol{AX}=\boldsymbol{b}$**，其中，**$\boldsymbol{A}\in\boldsymbol{R}^{n\times n}$ **为对称正定阵(且设** $\boldsymbol{A}$ **的特征值** $\lambda(\boldsymbol{A})$ **满足：**$0<\alpha\leqslant\lambda(\boldsymbol{A})\leqslant\beta$)**，则有迭代公式为**

$$\boldsymbol{X}^{(m+1)}=\boldsymbol{X}^{(m)}+\omega(\boldsymbol{b}-\boldsymbol{AX}^{(m)})\quad(m=0,1,2,\cdots)$$

证明：当 $0<\omega<\dfrac{2}{\beta}$ **时，上述迭代法收敛.**

证 由迭代公式 $\boldsymbol{X}^{(m+1)}=\boldsymbol{X}^{(m)}+\omega(\boldsymbol{b}-\boldsymbol{AX}^{(m)})$ 得

$$\boldsymbol{X}^{(m+1)}=(\boldsymbol{I}-\omega\boldsymbol{A})\boldsymbol{X}^{(m)}+\omega\boldsymbol{b}\overset{\text{记}}{=\!=}\boldsymbol{B}_\omega\boldsymbol{X}^{(m)}+\omega\boldsymbol{b}$$

其中，$\boldsymbol{B}_\omega=\boldsymbol{I}-\omega\boldsymbol{A}$，则由迭代法收敛的充要条件定理 5.1 知：当 $\rho(\boldsymbol{B}_\omega)=\max|1-\omega\lambda_i(\boldsymbol{A})|<1$ 时，上述迭代法收敛.($\lambda_i(\boldsymbol{A})$ 为 $\boldsymbol{A}$ 的任一特征值)，此即 $|1-\omega\lambda_i(\boldsymbol{A})|<1$，把此不等式展开有 $0<$

$\omega\lambda_i(\boldsymbol{A})<2$，由于 $\boldsymbol{A}$ 是对称正定阵，且其特征值满足 $0<\alpha\leqslant\lambda(\boldsymbol{A})\leqslant\beta$，又由设 $0<\omega<\dfrac{2}{\beta}$，于是有 $0<\omega\alpha\leqslant\omega\lambda(\boldsymbol{A})\leqslant\omega\beta<2$，即 $0<\omega\lambda(\boldsymbol{A})<2$ 成立. 所以，上述迭代法收敛.

例 17 设 A 是 n 阶按行严格对角占优阵，经过高斯消去法的第一步后，A 变为如下形式：$\begin{bmatrix} a_{11} & a_1^{\mathrm{T}} \\ 0 & \boldsymbol{A}_2 \end{bmatrix}$

证明：$\boldsymbol{A}_2$ 是 $n-1$ 阶的按行严格对角占优阵.

证 A 经过高斯消去法的第一步后，$\boldsymbol{A}_2$ 的元素为

$$a_{ij}^{(2)}=a_{ij}-\frac{a_{i1}}{a_{11}}a_{1j},(i,j=2,3,\cdots,n),$$

由于 A 是按行严格对角占优阵，可知有

$$|a_{ii}|>\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|,(i=1,2,\cdots,n) \tag{1}$$

当 $i=1$ 时，由式(1)得到：$|a_{11}|>\sum\limits_{j=2}^{n}|a_{1j}|$，两边乘以 $|a_{i1}|/|a_{11}|$，有 $|a_{i1}|>\sum\limits_{j=2}^{n}|a_{i1}|\left|\dfrac{a_{1j}}{a_{11}}\right|$，再移项得

$$0>-|a_{i1}|+\sum_{j=2}^{n}\left|\frac{a_{i1}}{a_{11}}a_{1j}\right|,\ (i=1,2,\cdots,n) \tag{2}$$

当 $i=2,3,\cdots,n$ 时，由式(1)和式(2)得

$$\begin{aligned}|a_{ii}|&>\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|-|a_{i1}|+\sum_{j=2}^{n}\left|\frac{a_{i1}}{a_{11}}\cdot a_{1j}\right|+\left|\frac{a_{i1}}{a_{11}}\cdot a_{1i}\right|\\&=\sum_{\substack{j=2\\j\neq i}}^{n}\left(|a_{ij}|+\left|\frac{a_{i1}}{a_{11}}\cdot a_{1j}\right|\right)+\left|\frac{a_{i1}}{a_{11}}\cdot a_{1i}\right|,\end{aligned}$$

故 $\left|a_{ii}-\dfrac{a_{i1}}{a_{11}}\cdot a_{1i}\right|\geqslant|a_{ii}|-\left|\dfrac{a_{i1}}{a_{11}}\cdot a_{1i}\right|$

$$>\sum_{\substack{j=2\\j\neq i}}^{n}\left(|a_{ij}|+\left|\frac{a_{i1}}{a_{11}}\cdot a_{1j}\right|\right)$$

$$\geqslant\sum_{\substack{j=2\\j\neq i}}^{n}\left|a_{ij}-\frac{a_{i1}}{a_{11}}\cdot a_{ij}\right|$$

即 $|a_{ii}^{(2)}|>\sum\limits_{\substack{j=2\\j\neq i}}^{n}|a_{ij}^{(2)}|$，$(i=2,\cdots,n)$

也即 A_2 是 $n-1$ 阶的按行严格对角占优阵.

例 18 设方阵 A 可逆，δA 为 A 的扰动矩阵，证明：当 $\|A^{-1}\delta A\|<1$ 时，方阵 $A+\delta A$ 也可逆.

证法 1 先证结论：若 n 阶方阵 B 满足 $\|B\|<1$，则 $I+B$ 为非奇异矩阵.

可用反证法证明. 若 $I+B$ 为奇异矩阵，则存在非零向量 X，满足齐次线性代数方程组 $(I+B)X=O$，则 $X=-BX$，取范数有 $\|X\|\leqslant\|B\|\ \|X\|$，注意到 $X\neq 0$，从而 $\|X\|>0$，不等式两边消去 $\|X\|$，即得 $\|B\|\geqslant 1$，这与条件 $\|B\|<1$ 矛盾，从而 $I+B$ 为非奇异矩阵.

现已知 $\|A^{-1}\delta A\|<1$，由上面结论得 $I+A^{-1}\delta A$ 为非奇异矩阵，又已知 A 可逆，即 A 也为非奇异矩阵，从而有

$A+\delta A=A(I+A^{-1}\delta A)$ 也是非奇异矩阵.

证法 2 因行列式 $|A+\delta A|=|A|\,|I+A^{-1}\delta A|$，由已知 A 可逆，即 $|A|\neq 0$，且 $\rho(A^{-1}\delta A)\leqslant\|A^{-1}\delta A\|<1$，所以 -1 不是 $A^{-1}\delta A$ 的特征值，即 $|I+A^{-1}\delta A|\neq 0$，从而有 $|A+\delta A|\neq 0$，即方阵 $A+\delta A$ 也可逆.

证法 3 反证法. 若 $A+\delta A$ 不可逆，则齐次线性方程组 $(A+\delta A)X=O$ 必有非零解 $X\neq 0$，对此 X，有 $AX+\delta AX=0$，因 A 可逆，所以有 $X=-A^{-1}\delta AX$，从而有 $\|X\|=\|-A^{-1}\delta AX\|\leqslant\|A^{-1}\delta A\|\ \|X\|$，因为 $X\neq 0$，所以 $\|X\|>0$，上式两边除以不为零的 $\|X\|$，得 $\|A^{-1}\delta A\|\geqslant 1$，这与假设 $\|A^{-1}\delta A\|<1$ 矛盾，所以

方阵 $A+\delta A$ 也可逆.

习　题　五

1. 利用(1)顺序高斯消去法;(2)紧凑格式求解方程组 $\boldsymbol{AX}=\boldsymbol{b}$,其中

$$\boldsymbol{A}=\begin{bmatrix}1&2&1&-2\\2&5&3&-2\\-2&-2&3&5\\1&3&2&3\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}4\\7\\-1\\0\end{bmatrix}.$$

2. 利用按列选主元素消去法求解方程组

$$\begin{cases}x_1+3x_2-2x_3-4x_4=3\\2x_1+6x_2-7x_3-10x_4=-2\\-x_1-x_2+5x_3+9x_4=14\\-3x_1-5x_2+15x_4=-6\end{cases}$$

3. 试用矩阵 A 的直接三角分解:$\boldsymbol{A}=\boldsymbol{LU}$,求解方程组

(1) $\begin{bmatrix}1&2&1\\2&2&3\\-1&-3&0\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}0\\3\\2\end{bmatrix}$;

(2) $\begin{bmatrix}1&2&3&4\\1&4&9&16\\1&8&27&64\\1&16&81&256\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}2\\10\\44\\190\end{bmatrix}$.

4. 试用矩阵的直接分解法,导出矩阵 A 的克洛脱分解:$A=\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{U}}$ 的计算公式,其中 $\widetilde{\boldsymbol{L}}$ 是下三角阵,$\widetilde{\boldsymbol{U}}$ 是单位上三角阵.

5. 证明

(1) 初等下三角阵

$$
\boldsymbol{L}_k = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & -l_{k+1,k} & 1 & & & \\ & & \vdots & & \ddots & & \\ & & -l_{n,k} & & & & 1 \end{bmatrix}
$$

的逆阵为 $\boldsymbol{L}_k^{-1} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & l_{k+1,k} & 1 & & & \\ & & \vdots & & \ddots & & \\ & & l_{n,k} & & & & 1 \end{bmatrix}$

（即只与 $\boldsymbol{L}_k$ 非对角元差一符号）.

（2）$\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$

6．已知

$$
\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 3 & 3 \\ 1 & 3 & 11 & 5 \\ 1 & 3 & 5 & 19 \end{bmatrix}
$$

试作 $\boldsymbol{A}$ 的乔列斯基分解：$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$，其中 $\boldsymbol{L}$ 是对角元全为正的下三角阵.

7．试用平方根法求解方程组

$$\begin{bmatrix} 4 & -2 & -4 \\ -2 & 17 & 10 \\ -4 & 10 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 3 \\ -7 \end{bmatrix}$$

8. **试用追赶法求解方程组**

$$\begin{cases} 2x_1 - x_2 = 1 \\ -x_1 + 2x_2 - x_3 = \dfrac{1}{2} \\ -x_2 + 2x_3 - x_4 = \dfrac{1}{3} \\ -x_3 + 2x_4 = \dfrac{1}{4} \end{cases}$$

9. **设线性方程组为**

$$\begin{cases} 7x_1 + 10x_2 = 1 \\ 5x_1 + 7x_2 = 0.7 \end{cases}$$

(1) **试求系数矩阵 $\boldsymbol{A}$ 的条件数 $\mathrm{Cond}(\boldsymbol{A})_\infty$；**

(2) **若右端向量有扰动 $\delta\boldsymbol{b} = (0.01, -0.01)^{\mathrm{T}}$，试估计解的相对误差.**

10. **分别用雅可比迭代法与赛德尔迭代法求解方程组**

$$\begin{cases} 20x_1 + 2x_2 + 3x_3 = 24 \\ x_1 + 8x_2 + x_3 = 12 \\ 2x_1 - 3x_2 + 15x_3 = 30 \end{cases}$$

取初值 $\boldsymbol{X}^{(0)} = (0,0,0)^{\mathrm{T}}$，精确到小数后四位，并要求分别写出其迭代法的分量形式和矩阵形式.

11. **对方程组**

$$\begin{cases} x_1 + 2x_2 - 2x_3 = 1 \\ x_1 + x_2 + x_3 = 2 \\ 2x_1 + 2x_2 + x_3 = 3 \end{cases}$$

分别讨论用 Jacobi 迭代法和 G-S 迭代法求解的收敛性.

12. 设方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 中

$$\boldsymbol{A}=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & 1 & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2} & 1\end{bmatrix}$$

分别讨论用 Jacobi 迭代法，G-S 迭代法和 SOR 迭代法（$0<\omega<2$）求解的收敛性.

13. 用下面的例子说明赛德尔迭代法的收敛、发散性质可能由方程组中方程或未知元编号的改变而改变.

(1) $\begin{cases}3x+\ y+2z=6\\ \quad\ 4y+\ z=8;\\ x\qquad +2z=2\end{cases}$　　(2) $\begin{cases}2z+\ y+3x=6\\ z+4y\qquad =8.\\ 2z\qquad +\ x=2\end{cases}$

14. 用逐次超松驰迭代法解方程组（取 $\omega=0.9$，$\boldsymbol{X}^{(0)}=(0,0,0)^{\mathrm{T}}$）

$$\begin{cases}5x_1+2x_2+\ x_3=-12\\ -x_1+4x_2+2x_3=20\\ 2x_1-3x_2+10x_3=3\end{cases}$$

精确到小数后四位.

15. 用下面的例子说明 5.5 节[判别条件 Ⅰ]对雅可比迭代法来说只是一个充分条件而不是必要条件：

$$\boldsymbol{A}=\begin{bmatrix}3 & 2 & 2\\ 0 & 2 & 1\\ 1 & 0 & 2\end{bmatrix}.$$

第六章　常微分方程初值问题的数值解法

在许多科学研究和工程技术中，所建立的数学模型都是常微分方程或常微分方程组，但是除了少数特殊类型的方程能用解析方法求得其精确解外，多数情况要找出解的解析表达式是极其困难的，甚至是不可能的. 因此，只能用近似方法求得其数值解. 利用计算机解常微分方程主要使用数值方法. 本章主要介绍一阶常微分方程初值问题

$$\begin{cases} \dfrac{dy}{dx}=f(x,y) & (6.1)\\ y(a)=y_0 & (6.2)\end{cases}$$

在区间 $a\leqslant x\leqslant b$ 上的数值解法.

在常微分方程理论中已经知道，如果方程式(6.1)的右端函数 $f(x,y)$在区域 $R:\{a\leqslant x\leqslant b,-\infty<y<+\infty\}$上连续，并且关于 y 满足李普希兹(Lipschitz)条件：

$$|f(x,y_1)-f(x,y_2)|\leqslant L|y_1-y_2|$$

其中，(x,y_1)，$(x,y_2)\in R$，$L>0$ 为李普希兹常数. 则理论上就可以保证初值问题式(6.1)和式(6.2)的解 $y=y(x)$在$[a,b]$上存在且唯一. 今后如无特殊说明，我们总假定方程式(6.1)和式(6.2)的解 $y(x)$是存在、唯一且足够光滑，此外还假定方程本身是稳定的.

所谓初值问题的数值解法，就是能算出精确解 $y(x)$在自变量 x 的一系列离散节点

$$a=x_0<x_1<x_2<\cdots<x_{n-1}<x_n<\cdots$$

上的近似解

$$y_1, y_2, \cdots, y_{n-1}, y_n, \cdots$$

的方法. 这里,把 $y_k(k=1,2,\cdots)$叫做初值问题在点列 x_k 上的数值解. 也即数值解法的基本思想是通过某种离散化方法,将微分方程转化为差分方程(代数方程)来求解.

最常用的离散方法是数值积分法及泰勒展开法.

相邻两个节点间的距离 $h=x_{k+1}-x_k$ 称作步长,为了讨论方便,下面如不加特殊说明,一般都把 h 取为定步长,这时节点可表为

$$x_k=x_0+kh, \quad (k=0,1,2,\cdots)$$

初值问题式(6.1)和式(6.2)的数值解法有个基本特点,它们都采取"步进式",即求解过程可顺着节点排列的次序一步一步地向前推进,描述这类算法,只要给出用已知结果 $y_n, y_{n-1}, y_{n-2}, \cdots$ 计算 y_{n+1}的递推公式.

本章主要介绍初值问题中的一类基本数值解法:单步法,这类方法在计算 y_{n+1}时,只用到前一步 x_n 上的信息 y_n,因此单步法的特点是"自开始"的,即只要已知初始值 y_0,就可以按计算公式顺次求出数值解 $y_1, y_2, \cdots$,如欧拉(Euler)方法,龙格-库塔(Runge-Kutta)方法等. 下面将讨论几种常用单步法的构造,并引进一些重要概念,如收敛性、稳定性等,并给出与之相关的理论问题. 对线性多步法,方程组和高阶方程的数值解法只作简单介绍.

6.1　欧拉(Euler)方法

一、欧拉公式及其几何意义

欧拉方法是求解初值问题式(6.1)的一类最简单的单步法.

我们看到式(6.1)中,方程$\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$含有导数项 $y'(x)$,这是微分方程的本质特征,也是它难以求解的困难所在. 数值解法的关键,在于设法消除其导数项,这一工作就称为“离散化”. 例如,由于差商是微商的近似运算,因此可用来实现其离散化.

用差商代替式(6.1)中的微商,即

$$y'(x)\approx\frac{y(x_{n+1})-y(x_n)}{x_{n+1}-x_n}$$

并以 y_n,y_{n+1} 分别代替 $y(x_n)$及 $y(x_{n+1})$,就得到

$$\frac{y_{n+1}-y_n}{h}=f(x_n,y_n)$$

即

$$y_{n+1}=y_n+hf(x_n,y_n),\quad (n=0,1,2,\cdots) \tag{6.3}$$

式(6.3)就是著名的欧拉公式,它是一个显式的差分方程,由于初值 y_0 已知,从式(6.3)可得

$$y_1=y_0+hf(x_0,y_0)$$

相继地可以求得 $y_2,y_3,\cdots,y_n,\cdots$. 利用欧拉公式求初值问题式(6.1)、式(6.2)的方法就称为欧拉方法.

欧拉方法的几何意义是十分清楚的,方程$\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$的解 $y=y(x)$是 xOy 平面上的一族积分曲线. 初值问题式(6.1)和式(6.2)的解,几何上表示过点(a,y_0)的一条特殊的积分曲线 $y=y(x)$,欧拉方法的求解过程是如下进行的(图 6-1):

先在初始点 $P_0(a,y_0)$作积分曲线 $y=f(x)$的切线(其斜率为 $f(a,y_0)$),记此切线与直线 $x=x_1$ 交点 P_1 的纵坐标为 y_1;然后过点 $P_1(x_1,y_1)$以 $f(x_1,y_1)$为斜率作一直线,记它与直线 $x=x_2$ 交点 P_2 的纵坐标为 y_2,…,如此继续下去,可得一折线 $\overline{P_0P_1\cdots P_n}$,容易验证,该折线各个顶点的纵坐标 $y_i(i=0,1,\cdots,n)$

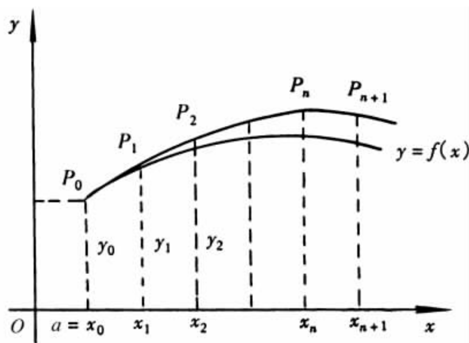


图 6-1

就是按欧拉公式(6.3)算得的数值解. 因此,欧拉方法又称为欧拉折线法. 显然,欧拉折线法简单地取切线的端点 P_n 作为折线的一个顶点,精度非常差,当步数增大时,由于误差的积累,用欧拉方法作出的折线 $\overline{P_0P_1\cdots P_n}$ 会越来越偏离积分曲线 $y=y(x)$.

二、梯形公式

将初值问题式(6.1)离散化的另一种常用方法是数值积分法. 为此,从另一角度来处理初值问题式(6.1)和式(6.2). 将方程(6.1)改写为

$$\mathrm{d}y(x)=f\,(x,y(x))\mathrm{d}x$$

再两边对 x 作在区间$[x_n, x_{n+1}]$上的积分,得

$$y(x_{n+1})=y(x_n)+\int_{x_n}^{x_{n+1}} f\,(x,y(x))\mathrm{d}x$$

其中 $y(x)$是初值问题的精确解.

要得到 $y(x_{n+1})$的值,就必须计算右端的积分:

$$\int_{x_n}^{x_{n+1}} f\,(x,y(x))\mathrm{d}x$$

但积分式中含有未知函数 $y(x)$,无法直接计算,若我们采用数值

积分方法计算积分 $\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x$ 的近似值,就可以获得 $y(x_{n+1})$ 的近似值. 这种将初值问题式(6.1)离散化的方法就称为数值积分法. 当选用不同的数值积分公式时,就会得到不同的计算公式.

例如,用最简单的左矩形公式计算积分值,得

$$\int_{x_n}^{x_{n+1}} f(x, y(x))\mathrm{d}x \approx hf(x_n, y(x_n))$$

于是就得到 $y(x)$ 在 $x=x_{n+1}$ 处的近似值

$$y(x_{n+1}) \approx y(x_n) + hf(x_n, y(x_n))$$

此式中,如用 y_n 代替右端的 $y(x_n)$,并将算出的右端值作为 $y(x_{n+1})$ 的近似值 y_{n+1},这样建立起来的计算公式,就是前述的欧拉公式:

$$y_{n+1} = y_n + hf(x_n, y_n), \quad (n=0,1,2,\cdots)$$

因矩形求积公式的精度较低,因此,所导出的欧拉公式的精度也较低. 为了提高精确度,我们改用梯形公式在 $[x_n, x_{n+1}]$ 上计算右端的积分,得

$$\int_{x_n}^{x_{n+1}} f[x, y(x)]\mathrm{d}x \approx \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

于是得

$$y(x_{n+1}) \approx y(x_n) + \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

以 y_n, y_{n+1} 分别代替 $y(x_n), y(x_{n+1})$,则得差分方程

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

$$(n=0,1,2,\cdots) \tag{6.4}$$

称式(6.4)为梯形公式，由于数值积分的梯形公式比矩形公式的精度高．因此，梯形公式(6.4)的精度比欧拉公式高．

必须指出，欧拉公式与梯形公式在计算上有明显的区别：欧拉公式的特点是可以由 y_n 直接计算 y_{n+1}，是一个直接计算公式，通常称这类公式为显式；而梯形公式中因它的右端也含有未知的 y_{n+1}，故不能直接得到 y_{n+1}，必须通过解方程才能求得 y_{n+1}，这类公式称为隐式．显式和隐式两类公式各有特点，考虑到数值稳定性等其他因素，人们有时需要用隐式公式，但从算法的角度看，显式则远比隐式方便．

三、截断误差

一般说，不论用什么数值方法，即使在初始值是准确的，且计算过程没有舍入误差的情况下，所得的数值解往往并不与初值问题的准确解完全符合，记

$$e_n = y(x_n) - y_n$$

表示数值方法在节点 x_n 处的截断误差，因为它纯粹是由于微分方程的离散化产生的，所以又叫做离散误差或方法误差．

人们常以泰勒展开为工具分析计算公式的误差．为简化分析，有时假定 y_n 为准确，即在 $y_n = y(x_n)$ 的前提下估计误差 $y(x_{n+1}) - y_{n+1}$，这种误差称为局部截断误差．如图 6-2 所示．图上设点 p_n 落在积分曲线 $y = y(x)$ 上，计算由 x_n 到 x_{n+1} 这一步方法引起的误差 ε_{n+1} 即为局部截断误差．为了便于区别，又常将 e_n 称为整体截断误差．在以后，我们将会看到，局部截断误差与整体截断误差有着密切的联系，也在一定程度上刻划了方法的精度．

定义 6.1 若某种微分方程数值解公式的局部截断误差为

$$\varepsilon_{n+1} = O(h^{p+1})$$

则称这种方法具有 p 阶精度或称为 p 阶方法．这里，p 为非负整数．通常 p 越大，h 越小，截断误差越小，数值方法越精确．

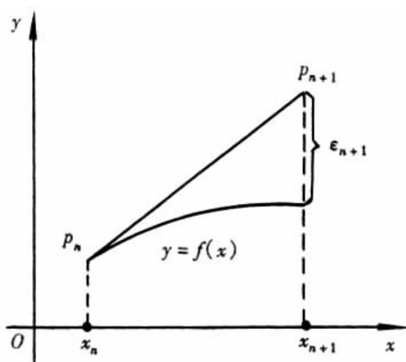


图 6-2

下面讨论欧拉公式和梯形公式的局部截断误差，有如下结论：

(1) 欧拉方法是一阶方法.

设
$$y_n = y(x_n)$$
把 $y(x_{n+1})$ 在 x_n 展开成泰勒级数，则有

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(\xi), \quad (x_n < \xi < x_{n+1})$$

用欧拉公式有

$$\begin{aligned} y_{n+1} &= y_n + hf(x_n, y_n) \\ &= y(x_n) + hf(x_n, y(x_n)) = y(x_n) + hy'(x_n) \end{aligned}$$

两式相减，得欧拉公式的局部截断误差为

$$T_1 = y(x_{n+1}) - y_{n+1} = \frac{h^2}{2}y''(\xi)$$

假定 $y(x)$ 在存在区间 $[a,b]$ 上充分光滑，并令

$$M = \max_{a<x<b} |y''(x)|$$

则
$$|T_1| \leqslant M \cdot \frac{h^2}{2} = O(h^2)$$

即欧拉方法的局部截断误差是 h^2 的同阶无穷小．正因为如此，所以常称欧拉方法具有一阶精度或称它是一阶方法．

（2）同样可证明，在 $y_n=y(x_n)$ 的前提下，梯形法的局部截断误差为

$$T_2=y(x_{n+1})-y_{n+1}=O(h^3)$$

即梯形法是二阶方法，比欧拉方法高一阶．

四、预测-校正法

梯形方法相对于欧拉方法虽然精度提高了，但是每一步都要用迭代法解方程，每迭代一次，都要重新计算函数 $f(x)$ 的值，而迭代又要反复进行若干次，计算量很大，且事先难以估计迭代次数．实际计算时，一种有效措施是构造所谓预测-校正法．当 h 较小时，使只迭代一二次就可转入下一步的计算，这就简化了算法．

例如：

$$\begin{cases}\text{预测} & \bar{y}_{n+1}=y_n+hf(x_n,y_n)\\ \text{校正} & y_{n+1}=y_n+\dfrac{h}{2}[f(x_n,y_n)+f(x_{n+1},\bar{y}_{n+1})]\end{cases} \quad (6.5)$$

就是一个预测-校正格式．具体地说，先用欧拉公式求得一个初步的近似值 $\bar{y}_{n+1}$，称之为预测值，预测值 $\bar{y}_{n+1}$ 的精度可能很差，再用梯形公式将它校正一次，即按式(6.5)迭代一次得 y_{n+1}，这个结果称校正值，作为初值问题式(6.1)和式(6.2)在节点 x_{n+1} 处的数值解．而这样建立的预测-校正格式通常也称为改进的欧拉公式．

一般，较为简单的预测-校正格式都包含两个计算公式：一个是显式公式，作为预测公式；另一个是隐式公式作为校正公式，当然也可以构造包含多个计算公式的预测-校正格式．构造预测-校正格式时，应该注意阶数的匹配，例如在式(6.5)中，校正公式具有二阶精度，而预测公式仅具有一阶精度，因为提供的预测值精度较差，且仅经一次校正，校正值的精度也不会太高．

下面的预测-校正格式：

$$\begin{cases} \text{预测} & \bar{y}_{n+1} = y_{n-1} + 2hf(x_n, y_n) \\ \text{校正} & y_{n+1} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})] \end{cases} \tag{6.6}$$

要好一些，因为可用泰勒展开的方法证得此预测公式也具有二阶精度，与校正公式是一致的．预测值精度提高了，可望校正值精度会更高一些．

式(6.6)中的预测公式

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$$

很容易用数值积分的中矩形公式推得(图 6-3)．

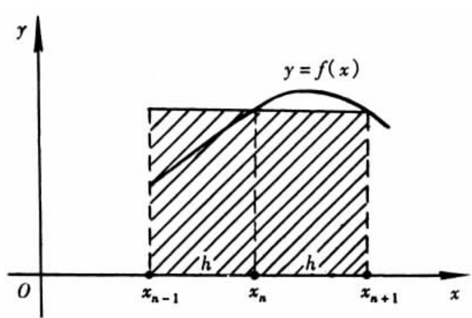


图 6-3

对方程

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y)$$

在区间$[x_{n-1}, x_{n+1}]$上积分，得

$$y(x_{n+1}) = y(x_{n-1}) + \int_{x_{n-1}}^{x_{n+1}} f(x, y(x))\mathrm{d}x$$

用中矩形公式计算积分

$$\int_{x_{n-1}}^{x_{n+1}} f(x, y(x))\mathrm{d}x \approx f(x_n, y(x_n))(x_{n+1} - x_{n-1})$$

分别用 y_{n-1}, y_n, y_{n+1} 代公式中 $y(x_{n-1}), y(x_n), y(x_{n+1})$ 即得

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) \tag{6.7}$$

对比欧拉公式、梯形公式及式(6.7),可发现,前面两个公式的共同点是,欲计算 y_{n+1},只用到前一步的 y_n,所以它们都是单步法. 而式(6.7)则不然,欲计算 y_{n+1},需要用到前两步的 y_{n-1} 及 y_n,称这样的方法为两步法. 两步法乃至更多步的方法称为多步法. 多步法的计算需用单步法提供足够的初始值,然后再能按计算公式计算其数值解.

例 6.1 分别用欧拉方法和改进的欧拉方法求解初值问题

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = y - \dfrac{2x}{y} \\ y(0) = 1 \end{cases}$$

在区间[0,1]上,步长 $h=0.1$ 的数值解.

解 此问题的精确解为 $y=\sqrt{1+2x}$,对此问题欧拉方法的具体计算公式为

$$\begin{cases} y_{n+1} = y_n + 0.1\left(y_n - \dfrac{2x_n}{y_n}\right) = 1.1y_n - \dfrac{0.2x_n}{y_n}, \ (n=0,1,2,\cdots,9) \\ y_0 = 1 \end{cases}$$

改进的欧拉方法的具体计算公式为

$$\begin{cases} \bar{y}_{n+1} = y_n + 0.1\left(y_n - \dfrac{2x_n}{y_n}\right) = 1.1y_n - \dfrac{0.2x_n}{y_n}, \ (n=0,1,2,\cdots,9) \\ y_{n+1} = y_n + \dfrac{0.1}{2}\left[\left(y_n - \dfrac{2x_n}{y_n}\right) + \left(\bar{y}_{n+1} - \dfrac{2x_{n+1}}{y_{n+1}}\right)\right] \end{cases}$$

由初值 $y_0=1$ 出发按上述公式分别计算,所得数值结果见表 6-1.

表 6-1 计算结果

x_n	y_n(欧拉法)	y_n(改进欧拉法)	精确解 $y(x_n)$
0	1	1	1
0.1	1.100000	1.095909	1.095445
0.2	1.191818	1.184096	1.183216
0.3	1.277438	1.266201	1.264911
0.4	1.358213	1.343360	1.341641
0.5	1.435133	1.416402	1.414214
0.6	1.508966	1.485956	1.483240
0.7	1.580338	1.552515	1.549193
0.8	1.649783	1.616476	1.612452
0.9	1.717779	1.678168	1.673320
1.0	1.784778	1.737869	1.732051

为了便于比较,表 6-1 中同时列出了解析解在各节点处具有七位有效数字的值 $y(x_n)$,从表中的数值可见,欧拉方法给出的数值解误差较大,而改进的欧拉方法明显地改善了精度.

6.2 龙格-库塔(Runge-Kutta)方法

龙格-库塔方法是一类应用较广的高精度的单步法,简称为 R-K 方法.

一、R-K 方法的基本思想

从前面欧拉方法的截断误差估计知,用一阶泰勒多项式近似函数可得其局部截断误差为一阶泰勒余项 $O(h^2)$. 完全类似地,若用 p 阶泰勒多项式近似函数,即

$$y_{n+1}=y(x_n)+hy'(x_n)+\frac{1}{2!}h^2y''(x_n)+\cdots+\frac{1}{p!}h^py^{(p)}(x_n)$$

其中，$y'(x)=f(x,y)$，$y''(x)=f'_x(x,y)+f'_y(x,y)f,\cdots$.

则局部截断误差应为 p 阶泰勒余项 $O(h^{p+1})$. 而提高泰勒公式的阶 p，即可提高计算结果的精度，从理论上讲，只要解 $y(x)$ 充分光滑，利用函数的泰勒展开可以构造任意高精度的数值方法. 但事实上，具体构造时往往是相当困难，因为若直接对 $y(x)$ 用高次泰勒多项式近似，则因公式中出现 $f(x,y)$ 的各阶偏导数，从而导致计算十分繁琐，工作量大而不实用. 因此，泰勒展开方法一般不直接使用，但可设法间接使用泰勒展开方法，以求得精度较高的数值方法. 下面介绍的 R-K 方法就是用间接泰勒展开方法来构造的.

为了导出 R-K 方法，先对前面介绍的欧拉公式与改进的欧拉公式作进一步的分析.

欧拉公式(6.3)可改写成
$$\begin{cases}y_{n+1}=y_n+hk_1\\k_1=f(x_n,y_n)\end{cases}\tag{6.8}$$

用它计算 y_{n+1}，需计算一次 $f(x,y)$ 的值，若设 $y(x_n)=y_n$，则 y_{n+1} 的表达式与 $y(x_{n+1})$ 的泰勒展开式的前两项完全相同，即局部截断误差为 $O(h^2)$.

而改进的欧拉公式(6.5)

$$\begin{cases}\bar{y}_{n+1}=y_n+hf(x_n,y_n)\\y_{n+1}=y_n+\dfrac{h}{2}[f(x_n,y_n)+f(x_{n+1},\bar{y}_{n+1})]\end{cases}$$

可改写为

$$\begin{cases}y_{n+1}=y_n+h\left(\dfrac{k_1}{2}+\dfrac{k_2}{2}\right)\\k_1=f(x_n,y_n)\\k_2=f(x_n+h,y_n+hk_1)\end{cases}\tag{6.9}$$

用它计算 y_{n+1}，需计算两次 $f(x,y)$ 的值，y_{n+1} 的展开式与 $y(x_{n+1})$ 的泰勒展开式的前三项完全相同，即局部截断误差为 $O(h^3)$.

上述两组公式在形式上有一个共同点：都是用 $f(x,y)$ 在某些点上的值的线性组合得出 $y(x_{n+1})$ 的近似值 y_{n+1}，而且可以看出，增加计算 $f(x,y)$ 的次数，可提高截断误差的阶.

类似地，龙格-库塔方法的基本思想也是设法计算 $f(x,y)$ 在某些点上的函数值，然后对这些函数值作线性组合，构造近似计算公式，再把近似公式和解的泰勒展开式相比较，使前面的若干项吻合，从而获得达到一定精度的数值计算公式.

具体构造时，先引进若干参数，例如，一般的显式 R-K 方法的形式为

$$\begin{cases} y_{n+1} = y_n + h\sum_{i=1}^{r}\omega_i k_i \\ k_1 = f(x_n, y_n) \\ k_i = f\left(x_n + \alpha_i h, y_n + h\sum_{j=1}^{i-1}\beta_{ij}k_j\right), (i=2,3,\cdots,r) \end{cases} \tag{6.10}$$

其中，参数 $\omega_i, \alpha_i, \beta_{ij}$ 是既与方程式(6.1)右端 $f(x,y)$ 无关，又与步数 n 无关的常数，选择它们的原则是要求式(6.10)的右端在 (x_n, y_n) 处作泰勒展开后，按 h 的幂次作升幂排列重新整理，得式

$$y_{n+1} = y_n + \gamma_1 h + \frac{1}{2!}\gamma_2 h^2 + \frac{1}{3!}\gamma_3 h^3 + \cdots \tag{6.11}$$

再与微分方程式(6.1)的精确解 $y(x)$ 在点 $x = x_n$ 处的泰勒展开式

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{1}{2!}h^2 y''(x_n) + \cdots$$

$$+ \frac{h^p}{p!}y^{(p)}(x_n) + O(h^{p+1})$$

相比较，使其有尽可能多的项重合，例如，要求

$$\gamma_1=f_n,\gamma_2=f'_n,\gamma_3=f''_n,\cdots,\gamma_p=f_n^{(p-1)}$$

就得到 p 个方程,从而定出参数 $\omega_i,\alpha_i,\beta_{ij}$,再代入 $k_1,k_2,\cdots,k_r$ 的表达式,就可得到计算微分方程初值问题式(6.1)、式(6.2)的数值计算公式:

$$y_{n+1}=y_n+h\sum_{i=1}^{r}\omega_i k_i \tag{6.10}$$

式(6.10)就称为 r 段的 R-K 方法的计算公式.

若式(6.11)与 $y(x_{n+1})$ 的泰勒展开式的前 $p+1$ 项完全一致,即使局部截断误差达到 $O(h^{p+1})$,则称公式(6.10)为 $\boldsymbol{p}$ 阶 $\boldsymbol{r}$ 段的 **R-K** 方法.

通常用得较多的是 r 取为 2,3,4,在这些情况下,为了计算方便,可适当选取 $\alpha_i,\beta_{ij},\omega_i$,使 $p=r$(即阶数=段数). 这样问题就归结为怎样选取参数 $\omega_i,\alpha_i,\beta_{ij}$,使式(6.10)成为所需精度的计算方法.

二、龙格-库塔公式的推导

现在就按上述介绍的龙格-库塔方法的基本思想,具体地推导一些常用的龙格-库塔公式.

由于选取 $\omega_i,\alpha_i,\beta_{ij}$ 的推导涉及复合函数求导和二元函数的泰勒展开,显然,公式的段数 r 越大,推导越繁琐,但基本思想方法一样. 为了说明清楚起见,下面仅以二阶二段龙格-库塔方法为例,讨论推导过程.

从 R-K 方法的一般式(6.10)得,当 $r=2$ 时,龙格-库塔格式为

$$\begin{cases}y_{n+1}=y_n+h(\omega_1k_1+\omega_2k_2)\\ k_1=f(x_n,y_n)\\ k_2=f(x_n+\alpha_2h,y_n+h\beta_{21}k_1)\end{cases} \tag{6.12}$$

适当选取参数 $\omega_1,\omega_2,\alpha_2,\beta_{21}$ 的值,使得在 $y(x_n)=y_n$ 的假设下,局

部截断误差 $y(x_{n+1})-y_{n+1}$ 的阶尽可能高. 若要使

$$y(x_{n+1})-y_{n+1}=O(h^3)$$

可将 $y_{n+1}=y_n+h(\omega_1 k_1+\omega_2 k_2)$ 在 (x_n,y_n) 处作泰勒展开,为此,把 k_1,k_2 的表示式代入式(6.12),得

$$y_{n+1}=y_n+\omega_1 hf(x_n,y_n)+\omega_2 hf(x_n+\alpha_2 h,y_n+\beta_{21}hf_n)$$

显然,上式中前二项不必展开,且

$$f(x_n,y_n)=f_n$$

只要把 $f(x_n+\alpha_2 h,y_n+\beta_{21}hf_n)$ 在 (x_n,y_n) 处展开即可,也即

$$f(x_n+\alpha_2 h,y_n+\beta_{21}hf_n)=f(x_n,y_n)+\alpha_2 h\frac{\partial f}{\partial x}$$

$$+\beta_{21}h\frac{\partial f}{\partial y}f_n+O(h^2)$$

其中,$\frac{\partial f}{\partial x}$,$\frac{\partial f}{\partial y}$ 都是在 x_n,y_n 处取值,记为 $\frac{\partial f_n}{\partial x}$,$\frac{\partial f_n}{\partial y}$,从而

$$\begin{aligned}y_{n+1}&=y_n+h(\omega_1 k_1+\omega_2 k_2)\\&=y_n+\omega_1 hf_n+\omega_2 h\left[f_n+h\left(\alpha_2\frac{\partial f_n}{\partial x}+\beta_{21}\frac{\partial f_n}{\partial y}f_n\right)\right]+O(h^3)\end{aligned}$$

再按 h 的幂次作升幂排列整理得

$$y_{n+1}=y_n+(\omega_1+\omega_2)f_n h+\omega_2\left(\alpha_2\frac{\partial f_n}{\partial x}+\beta_{21}\frac{\partial f_n}{\partial y}f_n\right)h^2+O(h^3)$$

与微分方程 $\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$ 的精确解 $y=y(x)$ 在点 $x=x_n$ 处的泰勒展开式

$$\begin{aligned}y(x_{n+1})&=y(x_n+h)\\&=y(x_n)+y'(x_n)h+\frac{1}{2!}y''(x_n)h^2+O(h^3)\end{aligned}$$

$$=y_n+f_nh+\frac{1}{2}\left(\frac{\partial f_n}{\partial x}+\frac{\partial f_n}{\partial y}f_n\right)h^2+O(h^3)$$

逐项进行比较,为使

$$y(x_{n+1})-y_{n+1}=O(h^3)$$

则令 h,h^2 **项的系数相同,便得到三个等式:**

$$\begin{cases}\omega_1+\omega_2=1\\ \omega_2\alpha_2=\dfrac{1}{2}\\ \omega_2\beta_{21}=\dfrac{1}{2}\end{cases}\tag{6.13}$$

将由这三个等式中确定的 $\omega_i,\alpha_i,\beta_{ij}$ **代入** R-K **计算式**(6.12),**即得二阶两段的** R-K **方法的计算格式. 由于上述方程组**(6.13)**是关于四个未知元** $\alpha_2,\beta_{21},\omega_1,\omega_2$ **的三个方程,因而有一个未知元可作为自由参数,这表明式**(6.13)**有无穷多组解,而且每一组解所构成的两段** R-K **方法的阶数都是** 2,**都叫做二阶两段的** R-K **公式,所以** R-K **公式是一类公式,每特殊确定一组系数,就得到一个特殊的** R-K **公式.**

例 6.2　若令 $\alpha_2=1$,**则从式**(6.13)**解得**

$$\omega_1=\omega_2=\frac{1}{2},\ \beta_{21}=1$$

则得

$$\begin{cases}y_{n+1}=y_n+h\left(\dfrac{1}{2}k_1+\dfrac{1}{2}k_2\right)\\ k_1=f(x_n,y_n)\\ k_2=f(x_n+h,y_n+hk_1)\end{cases}$$

这就是前面的预估-校正公式(6.9).

若取　$\omega_1=0,\omega_2=1,\alpha_2=\dfrac{1}{2},\beta_{21}=\dfrac{1}{2}$

则得另一个二阶两段的 R-K 计算公式为

$$\begin{cases} y_{n+1} = y_n + hk_2 \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}hk_1\right) \end{cases}$$

这也是常用的二阶公式，称为中点公式.

显然，还有许多可能取法，相应的计算公式的局部截断误差都是 $O(h^3)$，并由此可见，上节所讨论的欧拉方法及各种改进也可看成龙格-库塔方法的特殊情况.

可能有人会问，在上述二阶 R-K 计算格式中，若计算函数值的次数不增加，能否有其他选择参数的办法，使局部截断误差的阶再提高呢？譬如，能否适当选择这四个参数，使近似公式的局部截断误差的阶达到 $O(h^4)$ 呢？

事实上，因为在 $y(x_{n+1})$ 的展开式中的 $f_x f_y + f_y^2 f$ 是不能通过选择 $\alpha_2, \beta_{21}, \omega_1, \omega_2$ 来消掉，所以不论四个参数如何选择都不能再提高局部截断误差的阶，这说明在计算两次函数值的情况下，局部截断误差的阶最高是 $O(h^3)$，要进一步提高方法的阶，就必须通过继续增加计算函数值的次数的途径.

高阶龙格-库塔公式的推导方法与上面讨论的二阶 R-K 方法推导类似，只是随着阶数的增高，推导的工作量也随之增大.

例如，在 R-K 方法的一般式中，取 $r=3$，则有

$$\begin{cases} y_{n+1} = y_n + h(\omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3) \\ \text{其中} \quad k_1 = f(x_n, \quad y_n) \\ k_2 = f(x_n + \alpha_2 h, \quad y_n + h\beta_{21} k_1) \\ k_3 = f(x_n + \alpha_3 h, \quad y_n + h\beta_{31} k_1 + h\beta_{32} k_2) \end{cases} \tag{6.14}$$

仿照二阶情况的推导，只要把 k_1, k_2, k_3 代入 y_{n+1} 的表达式

中，在(x_n, y_n)处作泰勒展开，再与$y(x_{n+1})$在点x_n处的泰勒展开式比较，欲使公式的局部截断误差$y(x_{n+1})-y_{n+1}=O(h^4)$，可得含有八个参数$\omega_1, \omega_2, \omega_3, \alpha_2, \alpha_3, \beta_{21}, \beta_{31}, \beta_{32}$的六个方程组成的方程组：

$$\begin{cases} \omega_1+\omega_2+\omega_3=1 \\ \alpha_2=\beta_{21} \\ \alpha_3=\beta_{31}+\beta_{32} \\ \omega_2\alpha_2+\omega_3\alpha_3=\dfrac{1}{2} \\ \omega_2\alpha_2^2+\omega_3\alpha_3^2=\dfrac{1}{3} \\ \omega_2\beta_{32}\alpha_2=\dfrac{1}{6} \end{cases} \tag{6.15}$$

显然，有无穷多个解能满足方程组式(6.15)，而且每个这样的解所构成的三段龙格-库塔方法的阶数都是3，在这里不加推导而给出一个比较简单而重要的三阶龙格-库塔公式，称为库塔公式．即

$$\begin{cases} y_{n+1}=y_n+\dfrac{h}{6}(k_1+4k_2+k_3) \\ k_1=f(x_n,\quad y_n) \\ k_2=f\left(x_n+\dfrac{1}{2}h,\quad y_n+\dfrac{1}{2}hk_1\right) \\ k_3=f(x_n+h,\quad y_n-hk_1+2hk_2) \end{cases} \tag{6.16}$$

在实际应用中，最常用的是四阶龙格-库塔公式，但推导十分麻烦，要涉及到求解含有13个参数的11个方程组成的方程组，这里不加推导而仅给出两个常用的四阶龙格-库塔公式：

1．标准四阶龙格-库塔公式

$$
\begin{cases}
y_{n+1}=y_n+\dfrac{h}{6}(k_1+2k_2+2k_3+k_4)\\
k_1=f(x_n,\quad y_n)\\
k_2=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{1}{2}hk_1\right)\\
k_3=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{1}{2}hk_2\right)\\
k_4=f(x_n+h,\quad y_n+hk_3)
\end{cases}
\tag{6.17}
$$

2．吉尔(Gill)公式

$$
\begin{cases}
y_{n+1}=y_n+\dfrac{h}{6}[k_1+(2-\sqrt{2})k_2+(2+\sqrt{2})k_3+k_4]\\
k_1=f(x_n,\quad y_n)\\
k_2=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{1}{2}hk_1\right)\\
k_3=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{\sqrt{2}-1}{2}hk_1+\dfrac{2-\sqrt{2}}{2}hk_2\right)\\
k_4=f\left(x_n+h,\quad y_n-\dfrac{\sqrt{2}}{2}hk_2+\dfrac{2+\sqrt{2}}{2}hk_3\right)
\end{cases}
\tag{6.18}
$$

其局部截断误差均为 $O(h^5)$.

从理论上说,我们可构造任意高阶的计算方法,但须注意精度的阶数与计算函数值的次数之间的关系并非等量增加的,应该指出,四阶及四阶以下的龙格-库塔方法,每一步需调用函数 $f(x,y)$ 的次数与阶数一致,如二阶公式需调用两次 $f(x,y)$,而四阶公式需调用四次 $f(x,y)$,但对于更高阶的情形则不然,需要调用 $f(x,y)$ 的次数,远远大于该方法的阶数. 由于计算量较大,从而很少使用更高阶的龙格-库塔方法. 事实上,对于大量的实际问题,四阶的龙格-库塔方法已可满足对精度的要求.

例 6.3 取步长 $h=0.2$,从 $x=0$ 直到 $x=1$ 用标准四阶龙格-库塔方法式(6.17)求解初值问题

$$\begin{cases} \dfrac{dy}{dx} = y - \dfrac{2x}{y} \\ y(0) = 1 \end{cases}$$

解　已知

$$f(x,y) = y - \frac{2x}{y}, y(0) = 1, h = 0.2$$

在$[0,1]$**上求** $x = 0.2, 0.4, 0.6, 0.8, 1.0$ **上的数值解.**

只要将 f, h **代入标准的四阶龙格-库塔格式**(6.17),**得具体计算公式为**

$$\begin{cases} y_{n+1} = y_n + \dfrac{0.2}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = y_n - \dfrac{2x_n}{y_n} \\ k_2 = y_n + 0.1k_1 - \dfrac{2(x_n + 0.1)}{y_n + 0.1k_1} \\ k_3 = y_n + 0.1k_1 - \dfrac{2(x_n + 0.1)}{y_n + 0.1k_2} \\ k_4 = y_n + 0.2k_3 - \dfrac{2(x_n + 0.2)}{y_n + 0.2k_3}, \quad (n = 0, 1, 2, \cdots) \end{cases} \tag{6.19}$$

从 $n=0$ **开始,每算得四个** $k_i(i=1,2,3,4)$**的值代入式**(6.19)**得** y_1**,再由** $n=1$ **算得四个** k_i **值,再代入式**(6.19)**算得** y_2**,依次类推,得方程的精确解在节点** $x=0.2,0.4,0.6,0.8,1.0$ **上的数值解** y_1, y_2, y_3, y_4, y_5**,其计算结果见表** 6-2.

表 6-2　　　　　　　　　计算结果

n	x_n	y_n	n	x_n	y_n
0	0	1	3	0.6	1.48328
1	0.2	1.18323	4	0.8	1.61251
2	0.4	1.34167	5	1.0	1.73214

和例 6.1 的结果相比较,可以看出,虽然这里步长增大了一倍,但精度却比欧拉方法和改进的欧拉方法都好.

龙格-库塔方法在求解范围较大而精确度要求较高时是比较好的方法,它和欧拉方法和改进的欧拉方法一样可以直接从头算起,且为了达到同样精确度,它所用的步长可以比欧拉方法和改进的欧拉方法所用的步长大得多. 但值得指出的是,由于龙格-库塔方法的推导基于泰勒展开方法,因而它要求所求的解具有较好的光滑性质. 反之,如果解的光滑性差,那么,使用四阶龙格-库塔方法求得的数值解,其精度可能反而不如改进的欧拉方法. 因此,在实际计算时,应当针对问题的具体特点选择合适的算法.

三、步长的选择

怎样选取合适的步长,这在实际计算中是很重要的. 因为步长愈小,每步计算的截断误差就愈小;但在一定的求解范围内,需要完成的步数就愈多,这不但引起计算量的增大,而且步数的增加往往造成舍入误差的严重积累. 上面介绍的龙格-库塔方法是对定步长(即步长 h 为常数)而言的,但为了保证精确度,一种有效的措施是在计算过程中自动调整步长,即变步长技巧.

现以四阶龙格-库塔方法为例,讨论如何自动选择步长,使计算结果满足给定精度的要求.

设从节点 x_n 出发,先以 h 为步长,利用四阶 R-K 方法经过一步计算得 $y(x_{n+1})$ 的近似值,记为 $y_{n+1}^{(h)}$,由于公式的局部截断误差是 $O(h^5)$,故有

$$y(x_{n+1})-y_{n+1}^{(h)}\approx ch^5 \tag{6.20}$$

当 h 不大时,c 可近似地看作常数. 然后将步长 h 折半,即取 $h/2$ 为步长,从 x_n 出发经过两步计算求得 $y(x_{n+1})$ 的近似值,记为 $y_{n+1}^{(h/2)}$,每一步计算的局部截断误差为 $c(h/2)^5$,于是有

$$y(x_{n+1})-y_{n+1}^{(h/2)}\approx 2c\left(\frac{h}{2}\right)^5$$

把它与式(6.20)相比，可得

$$\frac{y(x_{n+1})-y_{n+1}^{(h/2)}}{y(x_{n+1})-y_{n+1}^{(h)}}\approx\frac{1}{16}$$

经整理可得 $$y(x_{n+1})-y_{n+1}^{(h/2)}\approx\frac{1}{15}\left[y_{n+1}^{(h/2)}-y_{n+1}^{(h)}\right]$$

这表明以 $y_{n+1}^{(h/2)}$ 作为 $y(x_{n+1})$ 的近似值，其误差可用先后两次计算结果之差来表示. 因而，只需考察

$$|y_{n+1}^{(h/2)}-y_{n+1}^{(h)}|<\varepsilon$$

是否成立. 若成立，则可将 $y_{n+1}^{(h/2)}$ 作为 $y(x_{n+1})$ 的近似值；若不成立，则将步长再次折半进行计算，直到不等式成立为止，并取最后的 $y_{n+1}^{(h/2)}$ 作为计算结果. 以上方法就是计算过程中自动选择步长的方法，也称为变步长方法. 从表面上看，为了选择适当的步长，每一步的计算量增加了，但从总体考虑，工作量往往还是减少的.

龙格-库塔方法的主要优点是精确度较高，能满足通常的计算要求；容易编制程序；每次计算 y_{n+1}，只用到前一步的计算结果 y_n，因此，在已知初始值 y_0 的条件下，就可自动地进行计算(即"自开始"的)，是单步法，而且计算过程中可随时改变步长. 它的缺点是每前进一步需要计算多次 $f(x,y)$ 的值，因此，计算工作量较大，且其截断误差难以估计. 在实用上，一般，当要求更高的精确度时，采用的办法是缩小步长，而不是采用更高阶的公式，因为高阶公式的计算太复杂，一般选用标准四阶龙格-库塔方法即可.

6.3 收敛性与稳定性

收敛性与稳定性从不同的角度描述了数值方法的可靠性，只

有既收敛又稳定的方法，才能提供比较可靠的计算结果．下面只讨论单步法的收敛性与稳定性．

一般的显式单步法的计算公式可以统一地写成如下形式：

$$y_{n+1}=y_n+h\phi(x_n,y_n,h) \tag{6.21}$$

其中，h 为步长，$x_n=a+nh$，y_n 为方程的精确解 $y(x)$ 在点 x_n 处的近似值，而 $\phi(x,y,h)$ 称为方法的增量函数．它依赖于方程式(6.1)的右端 $f(x,y)$，且是关于所含变量的连续函数．

不同的单步法，对应于不同的增量函数．例如，欧拉方法的增量函数 $\phi(x,y,h)=f(x,y)$．

一、单步法的收敛性

微分方程初值问题的数值解法的基本思想是通过某种离散化手续，将微分方程转化为差分方程，用不同的离散化方法，便得到不同的单步法公式．这里，首先一个问题是，这种离散化是否合理，也即差分方程(6.21)的解当 $h\to 0$ 时（同时 $n\to\infty$）是否会收敛到微分方程的精确解．

定义 6.2 若某数值方法对任意固定的节点 $x_n=x_0+nh$，$(n=0,1,2,\cdots)$，当 $h\to 0$（同时 $n\to\infty$）时，有 $y_n\to y(x_n)$，则称该方法是收敛的．

由此可见，数值方法的收敛性并不涉及计算过程的舍入误差，而只与方法的截断误差有关．下面给出关于估计方法的截断误差及判别单步法收敛的充分定理．

定理 6.1 若(1)单步法 $y_{n+1}=y_n+h\phi(x_n,y_n,h)$ 中的增量函数 $\phi(x,y,h)$ 在区域 $a\leqslant x\leqslant b$，$-\infty<y<+\infty$，$0\leqslant h\leqslant h_0$ 上连续，并且关于 y 满足李普希兹条件：

$$|\phi(x,y_1,h)-\phi(x,y_2,h)|\leqslant L|y_1-y_2|,\ (L>0);$$

(2) 方法的局部截断误差 $\varepsilon_n=O(h^{p+1})$，　$(n=1,2,\cdots)$；

（3）初始值是精确的，即 $y(a)=y_0$

则　（1）方法的整体截断误差为 $e_n=O(h^p)$；

（2）当 $p\geqslant 1$ 时，方法是收敛的.

（证明略）

以上定理表明，只要连续的增量函数 $\phi(x,y,h)$ 关于 y 满足李普希兹条件，当 $e_0=y_0-y(a)=0$，且阶数 $p\geqslant 1$ 时，数值方法是收敛的，且若其局部截断误差 $\varepsilon_n=O(h^{p+1})$ 时，则其整体截断误差 $e_n=O(h^p)$，由于整体截断误差比局部截断误差低一阶，这一结论带有一般性，因此，在构造高精度的计算方法时，只要设法提高方法的局部截断误差即可.

依据定理 6.1，在初始值精确的前提下，判断单步法的收敛性，只要验证它们的增量函数 $\phi(x,y,h)$ 关于 y 能否满足李普希兹条件就可以了.

例如，对欧拉方法：$y_{n+1}=y_n+hf(x_n,y_n)$，由于其增量函数就是方程右端 $f(x,y)$，所以，当初值问题的右端 $f(x,y)$ 关于 y 满足李普希兹条件且初值 y_0 是精确时，它是收敛的.

类似地，不难验证其他单步法的收敛性.

例如，也可证明当方程的右端函数 $f(x,y)$ 在其定义区域上关于 y 满足李普希兹条件时，标准四阶 R-K 方法的增量函数关于 y 也满足李普希兹条件，从而这个方法也是收敛的.

二、单步法的稳定性

这里讨论的稳定性，不是初值问题本身的稳定性，而是指数值方法的稳定性，即数值稳定性. 即使是一个收敛的方法，在实际计算时，由于初始值一般都带有误差，同时在计算过程中一般总会产生舍入误差，且这些误差又必然会传播下去，对以后的计算结果产生影响. 数值稳定性问题讨论的就是指这种误差的积累能否得到控制的问题. 粗略地说，如果计算结果对初始数据的误差及计算

过程中的误差不敏感,也即舍入误差不增长,则称相应的数值方法是稳定的,否则,就称为不稳定.

数值稳定性有各种各样的定义,下面只简单地介绍"绝对稳定性"概念.

定义 6.3 设用某一数值方法计算 y_n 时,所得到的实际计算结果为 $\tilde{y}_n$,且由误差 $\delta_n = y_n - \tilde{y}_n$ 引起以后各节点处 $y_m(m>n)$ 的误差为 δ_m,如果总有 $|\delta_m| \leqslant |\delta_n|$,则称该数值方法是绝对稳定的.

数值稳定性的分析相当复杂,它不仅与方法本身有关,而且总跟方程的右端 $f(x,y)$ 以及步长有关,所以,对一般的微分方程,研究这个问题很困难. 但事实上,研究数值方法是否稳定,不可能也不必对每个不同的右端函数 $f(x,y)$ 进行讨论,为简单起见,通常只对试验方程(也称为模型方程)

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \lambda y \tag{6.22}$$

(其中 λ 为常数,当 λ 是复数时,$Re(\lambda)<0$)进行讨论,即研究将数值方法用于解方程式(6.22)得到的差分方程是否数值稳定. 这种思想来源于如下判断:若一个数值方法对如此简单问题都不是绝对稳定的话,那么,就难于用此数值方法来解一般的微分方程. 不言而喻,某一数值方法对试验方程是绝对稳定的,对一般方程不一定也是绝对稳定的,但试验方程在一定程度上还是反映了数值方法的特性.

在把某一数值方法应用于试验方程,为了保证数值稳定,步长 h 和 λ 都要受到一定的限制,故有定义:

定义 6.4 一个数值方法用于解试验方程式(6.22),若在 $\mu = \lambda h$ 平面中的某个区域 R 中方法都是绝对稳定的,而在域 R 外,方法是不稳定的,则称区域 R 是该数值方法的绝对稳定域.

显然,绝对稳定性区域越大,该方法的绝对稳定性越好,若某个方法的绝对稳定性区域包含 λh 复平面的左半平面,则称此数值

方法是 A-稳定的.

例如,对欧拉法:

$$y_{n+1}=y_n+hf(x_n,y_n),\quad (n=0,1,2,\cdots) \tag{6.23}$$

用它解试验方程(6.22)得

$$y_{n+1}=y_n+\lambda h y_n=(1+\lambda h)y_n,\quad (n=0,1,\cdots),$$

若 y_n 的实际计算值为 $\tilde{y}_n$,则由误差 $\delta_n=y_n-\tilde{y}_n$ 引起 y_{n+1} 的误差为 $\delta_{n+1}=(1+\lambda h)\delta_n$,故要对一切 n,总有 $|\delta_m|<|\delta_n|(m>n)$,只要取 $|1+\lambda h|<1$ 时,欧拉方法是绝对稳定的.

若记 $\mu=\lambda h$,即当 $|1+\mu|<1$ 时,欧拉方法是绝对稳定的. 当 $\lambda<0$时,可得 $0<h<-\dfrac{2}{\lambda}$时,欧拉方法绝对稳定;当 λ 为复数时,在 $\mu=\lambda h$ 的复平面上 $|1+\mu|<1$ 表示以 $(-1,0)$为圆心,1 为半径的单位圆内欧拉方法绝对稳定(图6-4). 显然,欧拉方法是条件稳定的.

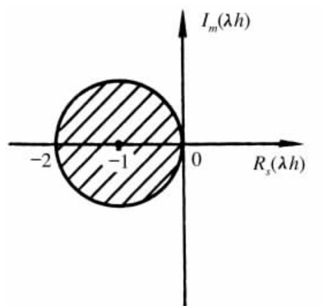


图 6-4 欧拉方法的绝对稳定区域

对于后退的欧拉方法:

$$y_{n+1}=y_n+hf(x_{n+1},y_{n+1}) \tag{6.24}$$

用它解试验方程(6.22)得 $y_{n+1}=y_n+\lambda h y_{n+1}$,解出 y_{n+1} 有 $y_{n+1}=\dfrac{1}{1-\lambda h}y_n$,则对于后退的欧拉方法,$\delta_n$ 应满足 $\delta_{n+1}=\dfrac{1}{1-\lambda h}\delta_n$,可见,它的绝对稳定性区域是 $|1-\mu|>1$,在 $\mu=\lambda h$ 的复平面上,这是以 1 为半径、$(1,0)$为圆心的圆外部,如图 6-5 所示. 由定义,该方法是 A-稳定的. 由图 6-4 和图 6-5 可见,后退欧拉方法的绝对稳定性区域比欧拉方法大得多,但可证明两种方法的阶数相同,只是欧拉方法是显式法,而后退欧拉方法是隐式法. 这说明隐式法的绝

对稳定性一般比同阶的显式法好得多.

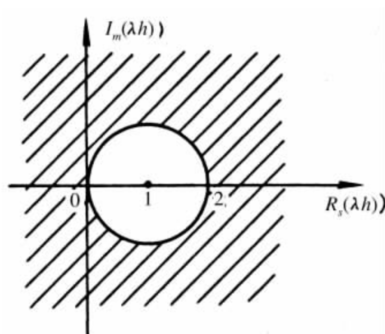


图 6-5　后退欧拉方法的绝对稳定区域

对二阶 R-K 法式(6.5),用于解试验方程式(6.22)得

$$y_{n+1}=y_n+\frac{1}{2}[\lambda h y_n+\lambda h(y_n+\lambda h y_n)]=\left[1+\lambda h+\frac{(\lambda h)^2}{2}\right]y_n,$$

令 $\mu=\lambda h$,则得 $$y_{n+1}=\left(1+\mu+\frac{1}{2}\mu^2\right)y_n$$

显然,当 $\left|1+\mu+\frac{\mu^2}{2}\right|<1$ 时,方法绝对稳定,这时方法的绝对稳定域是由曲线 $\left|1+\mu+\frac{\mu^2}{2}\right|=1$ 围成的区域内部.

用类似的方法对于"标准的四阶 R-K 方法"可推得

$$y_{n+1}=\left[1+\lambda h+\frac{1}{2}(\lambda h)^2+\frac{1}{6}(\lambda h)^3+\frac{1}{24}(\lambda h)^4\right]y_n$$

所以,其绝对稳定性区域为 $\left|1+\mu+\frac{\mu^2}{2!}+\frac{\mu^3}{3!}+\frac{\mu^4}{4!}\right|<1$. 若设 $\lambda<0$,由上式可近似地得到当 $0<h<-\frac{2.78}{\lambda}$ 时,四阶 R-K 方法绝对稳定.

收敛性和稳定性是两个重要的概念,在数值计算的不同分枝,它们的含义可以不同,这里介绍的收敛性反映数值公式本身的截

断误差对计算结果的影响;稳定性是反映某一计算步骤中出现的舍入误差对计算结果的影响. 稳定性是和步长 h 密切相关的,对于一种步长是稳定的数值公式,若将步长改大可能就不稳定,只有既收敛又稳定的数值方法才可以在实际计算中放心地使用. 因此,选择步长时不但要考虑到截断误差,还应考虑到绝对稳定性. 原则上是在满足稳定性及截断误差要求的前提下,步长尽可能取大一些.

6.4 线性多步法简介

单步法在计算 y_{n+1} 时,只用到前一步的信息 y_n. 为了提高精度,每步就要增加计算多个非节点处的函数值,如 R-K 方法,故计算量较大. 事实上,在计算 y_{n+1} 之前已经求得了一系列的近似值,如 $y_n, y_{n-1}, \cdots, y_{n-r}$ 等等,如果能充分利用这些信息来计算 y_{n+1},数值计算的结果可望具有较高的精确度,这就是多步法的基本思想. 多步法中最常用的是线性多步法,它的计算式中只出现 y_{n+1},$y_n, \cdots, y_{n-r}$ 及 $f(x_{n+1}, y_{n+1}), \cdots, f(x_{n-r}, y_{n-r})$ 的一次项,其一般形式为

$$\begin{aligned} y_{n+1} = {} & \alpha_0 y_n + \alpha_1 y_{n-1} + \cdots + \alpha_r y_{n-r} \\ & + h(\beta_{-1} f_{n+1} + \beta_0 f_n + \beta_1 f_{n-1} + \cdots + \beta_r f_{n-r}) \end{aligned} \tag{6.25}$$

可简写为 $$y_{n+1} = \sum_{k=0}^{r} \alpha_k y_{n-k} + h \sum_{k=-1}^{r} \beta_k f_{n-k} \tag{6.26}$$

这里 $f_j = f(x_j, y_j)$, $(j = n-r, n-r+1, \cdots, n+1)$, α_k, β_k 均为常数. 若 $\beta_{-1} \neq 0$,则式(6.25)是隐式;如 $\beta_{-1} = 0$,则式(6.25)是显式.

构造线性多步法公式常用泰勒展开方法和数值积分的方法,其中,泰勒展开方法应用更广. 可以说,所有形如式(6.25)的线性多步法的计算公式都可以利用泰勒展开方法构造出来,另外,它不仅可以得到求数值解的公式,而且容易估计截断误差. 下面仅对

其构造方法作简单介绍.

一、用数值积分的构造方法

考虑初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=f(x,y) & (6.1)\\ y(a)=y_0 & (6.2)\end{cases}$$

将方程式(6.1)的两端从 x_n 到 x_{n+1} 求积分,得积分方程

$$y(x_{n+1})=y(x_n)+\int_{x_n}^{x_{n+1}} f(x,y(x))\mathrm{d}x \tag{6.27}$$

为要通过这个积分关系式,获得 $y(x_{n+1})$ 的近似值,只要近似地算出其中的积分项 $\int_{x_n}^{x_{n+1}} f(x,y(x))\mathrm{d}x$,而选用不同的数值方法计算这个积分项,就会导出不同的计算公式. 这就是用数值积分的方法构造线性多步法的计算公式的基本思想. 例如,选取节点 $x_{n-3},x_{n-2},x_{n-1},x_n$ 及其对应的函数近似值 $y_{n-3},y_{n-2},y_{n-1},y_n$ 构造被积函数 $f(x,y(x))$ 的三次插值多项式 $p_3(x)$,用它近似代替式(6.27)中的被积函数 $f(x,y(x))$,并用 y_n 和 y_{n+1} 分别近似代替 $y(x_n)$和 $y(x_{n+1})$,则得

$$y_{n+1}=y_n+\int_{x_n}^{x_{n+1}} p_3(x)\mathrm{d}x \tag{6.28}$$

经计算积分后可得公式:

$$y_{n+1}=y_n+\frac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}) \tag{6.29}$$

这就是常用的阿达姆斯(Adams)四步外推显式公式,并由插值的余项公式 $R_3(x)$可推导出式(6.29)的局部截断误差为

$$R=\int_{x_n}^{x_{n+1}} R_3(x)\mathrm{d}x=\frac{251}{720}h^5 y^{(5)}(\eta)$$

$\eta\in(x_{n-3},x_{n+1})$，h 为步长．所以式(6.29)是四阶公式，故又称式(6.29)为四阶阿达姆斯外推公式．

若取节点 $x_{n-2},x_{n-1},x_n,x_{n+1}$ 为插值节点，用上述类似的方法可得四阶的阿达姆斯隐式公式：

$$y_{n+1}=y_n+\frac{h}{24}(9f_{n+1}+19f_n-5f_{n-1}+f_{n-2}) \qquad (6.30)$$

其局部截断误差为

$$R=-\frac{19}{720}h^5 y^{(5)}(\eta),\quad \eta\in(x_{n-2},x_{n+1})$$

由于积分区间在插值区间$[x_{n-2},x_{n+1}]$内，故阿达姆斯隐式公式又称为阿达姆斯内插公式．它是三步隐式公式．

由阿达姆斯计算公式(6.29)与公式(6.30)可知，阿达姆斯方法每一步仅调用一次函数 $f(x,y)$，因而计算量比 R-K 方法少．实际计算时常常把阿达姆斯外推公式和内插公式联合使用，构成阿达姆斯预测-校正系统，经常使用的是 $r=3$ 的情形，即

$$\begin{cases}\text{预测}\quad \bar{y}_{n+1}=y_n+\dfrac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3})\\ \text{校正}\quad y_{n+1}=y_n+\dfrac{h}{24}[9f(x_{n+1},\bar{y}_{n+1})+19f_n-5f_{n-1}+f_{n-2}]\end{cases} \qquad (6.31)$$

这是一个四步四阶公式，至于初始值除 y_0 以外，其他的 y_1,y_2,y_3 通常用四阶 R-K 公式计算．

二、用泰勒展开的构造方法

利用泰勒展开的方法构造线性多步法的计算公式的基本思想类同于 R-K 方法的推导，它是先将线性多步法公式(6.25)的右端

均在 $x=x_n$ 处作泰勒展开，然后按 h 的幂次作升幂排列重新整理得

$$y_{n+1}=(\sum_{k=0}^{r}\alpha_k)y_n+\sum_{j=1}^{p}\frac{h^j}{j!}[\sum_{k=1}^{r}(-k)^j\alpha_k+j\sum_{k=-1}^{r}(-k)^{j-1}\beta_k]y_n^{(j)}$$

$$+\frac{h^{p+1}}{(p+1)!}[\sum_{k=1}^{r}(-k)^{p+1}\alpha_k+(p+1)\sum_{k=-1}^{r}(-k)^p\beta_k]y_n^{p+1}+\cdots \tag{6.32}$$

于是，欲使公式(6.25)为 p 阶的，即局部截断误差为 $O(h^{p+1})$，只要令展开式(6.32)与 $y(x_{n+1})$在 x_n 处的泰勒展开式

$$y(x_{n+1})=\sum_{j=0}^{p}\frac{h^j}{j!}y_n^{(j)}+\frac{h^{p+1}}{(p+1)!}y_n^{(p+1)}+\cdots \tag{6.33}$$

相比较，能符合到 h^p 项，为此令

$$\begin{cases}\sum_{k=0}^{r}\alpha_k=1\\ \sum_{k=1}^{r}(-k)^j\alpha_k+j\sum_{k=1}^{r}(-k)^{j-1}\beta_k=1\end{cases} \tag{6.34}$$

$$(j=1,2,\cdots,p)$$

解出系数 α_k,β_k 代入式(6.25)，即得 p 阶的线性多步法公式.式(6.33)与式(6.32)相减可得方法的局部截断误差为

$$y(x_{n+1})-y_{n+1}=\frac{h^{p+1}}{(p+1)!}[1-\sum_{k=1}^{r}(-k)^{p+1}\alpha_k$$

$$-(p+1)\sum_{k=-1}^{r}(-k)^p\beta_k]y_n^{(p+1)}+\cdots \tag{6.35}$$

例 6.4 构造一个以 x_{n-1},x_n,x_{n+1} 为节点的线性两步公式

$$y_{n+1}=\alpha_0 y_n+\alpha_1 y_{n-1}+h(\beta_{-1}f_{n+1}+\beta_0 f_n+\beta_1 f_{n-1}) \tag{6.36}$$

解 显然，关键只要确定公式中的系数 $\alpha_0,\alpha_1,\beta_{-1},\beta_0,\beta_1$. 假

设前几步计算结果都是准确的，即 $y_i=y(x_i)$，$y'(x_i)=f(x_i,y_i)$，$(i\leqslant n)$，并记 $y_n^{(k)}=y^{(k)}(x_n)$，$(k=1,2,\cdots)$，将式(6.36)的右端均在 $x=x_n$ 处作泰勒展开有

$$y_{n-1}=y(x_n-h)$$

$$=y_n-y_n'h+\frac{1}{2!}y_n''h^2-\frac{1}{3!}y_n'''h^3+\frac{1}{4!}y_n^{(4)}h^4-\frac{1}{5!}y_n^{(5)}h^5+O(h^6)$$

$$f_{n+1}=f(x_{n+1},y_{n+1})\approx y'(x_{n+1})$$

$$=y_n'+y''h+\frac{1}{2!}y_n'''h^2+\frac{1}{3!}y_n^{(4)}h^3+\frac{1}{4!}y_n^{(5)}h^4+O(h^5)$$

$$f_n=f(x_n,y_n)=y_n'$$

$$f_{n-1}=f(x_{n-1},y_{n-1})=y'(x_{n-1})$$

$$=y_n'-y_n''h+\frac{1}{2!}y_n'''h^2-\frac{1}{3!}y_n^{(4)}h^3+\frac{1}{4!}y_n^{(5)}h^4+O(h^5)$$

将以上各式代入式(6.36)，并按 h 的升幂排列整理得

$$y_{n+1}=(\alpha_0+\alpha_1)y_n+(-\alpha_1+\beta_{-1}+\beta_0+\beta_1)y_n'h+\left(\frac{\alpha_1}{2}+\beta_{-1}-\beta_1\right)y_n''h^2$$

$$+\left(-\frac{\alpha_1}{6}+\frac{\beta_{-1}}{2}+\frac{\beta_1}{2}\right)y_n'''h^3+\left(\frac{\alpha_1}{24}+\frac{\beta_{-1}}{6}-\frac{\beta_1}{6}\right)y_n^{(4)}h^4$$

$$+\left(-\frac{\alpha_1}{120}+\frac{\beta_{-1}}{24}+\frac{\beta_1}{24}\right)y_n^{(5)}h^5+O(h^6) \tag{6.37}$$

为使公式(6.36)有 p 阶精度，只需使式(6.37)与 $y(x_{n+1})$ 在 x_n 处的泰勒展开式

$$y(x_{n+1})=y_n+y_n'h+\frac{1}{2!}y_n''h^2+\frac{1}{3!}y_n'''h^3+\cdots$$

的前 $p+1$ 项关于 h 的同次幂系数完全相同，若令 $p=4$，可得方程组：

$$
\begin{cases}
\alpha_0+\alpha_1=1\\
-\alpha_1+\beta_{-1}+\beta_0+\beta_1=1\\
\frac{1}{2}\alpha_1+\beta_{-1}-\beta_1=\frac{1}{2}\\
-\frac{1}{6}\alpha_1+\frac{1}{2}\beta_{-1}+\frac{1}{2}\beta_1=\frac{1}{6}\\
\frac{1}{24}\alpha_1+\frac{1}{6}\beta_{-1}-\frac{1}{6}\beta_1=\frac{1}{24}
\end{cases}
\tag{6.38}
$$

从而可解得 $\alpha_0=0,\alpha_1=1,\beta_{-1}=\frac{1}{3},\beta_0=\frac{4}{3},\beta_1=\frac{1}{3}$

代入式(6.36)得到计算公式：

$$
y_{n+1}=y_{n-1}+\frac{h}{3}(f_{n+1}+4f_n+f_{n-1}) \tag{6.39}
$$

称为辛甫生公式，显然，它是四阶公式，局部截断误差为

$$
R=-\frac{1}{90}h^5y_n^{(5)}+O(h^6)
$$

式(6.39)也可用数值积分方法得到.

当然，也可以只要求式(6.38)中前面几个方程成立，例如，只要求前面三个方程成立，而从前面三个方程中解 5 个待定系数，则有两个自由参数，如令 $\alpha_0=\beta_{-1}=0$，可解出 $\alpha_1=1,\beta_1=0,\beta_0=2$，代入式(6.36)得计算公式：

$$
y_{n+1}=y_{n-1}+2hf(x_n,y_n)
$$

此即在 6.1 节中用数值积分方法得到的公式(6.7)，显然它也是一个线性两步公式，但它只是一个二阶公式.

类似地，用泰勒展开方法还可以构造常用的米尔尼(Milne)公式：

$$
y_{n+1}=y_{n-3}+\frac{4}{3}h(2f_{n-2}-f_{n-1}+2f_n) \tag{6.40}
$$

这是一个四阶四步的显式线性公式,其局部截断误差为

$$R=\frac{14}{45}h^5 y_n^{(5)}(\xi)$$

又可构造稳定性较好的哈明(Hamming)公式:

$$y_{n+1}=\frac{1}{8}[9y_n-y_{n-2}+3h(f_{n+1}+2f_n-f_{n-1})] \tag{6.41}$$

此是一个四阶三步的隐式线性公式,这个公式不能用数值积分方法推得,其局部截断误差为

$$R=-\frac{1}{40}h^5 y^{(5)}(\xi)$$

6.5 一阶常微分方程组和高阶方程

前几节中,我们讨论了一阶常微分方程初值问题的数值解法.只要引进向量记号,把 y 和 f 都理解为向量,那么,前面所介绍的各种数值计算公式都可平行地推广到一阶常微分方程组的情况.

一、一阶常微分方程组

考虑一阶常微分方程组的初值问题:

$$\begin{cases} y_i'(x)=f_i(x,y_1(x),y_2(x),\cdots,y_m(x)) \\ y_i(a)=y_{i0} \end{cases} \tag{6.42}$$

$$(i=1,2,\cdots,m)$$

引进向量记号:

$$\boldsymbol{Y}(x)=(y_1(x),y_2(x),\cdots,y_m(x))^{\mathrm{T}},$$

$$\boldsymbol{f}(x,\boldsymbol{Y})=(f_1(x,\boldsymbol{Y}),f_2(x,\boldsymbol{Y}),\cdots,f_m(x,\boldsymbol{Y}))^{\mathrm{T}}$$

$$\boldsymbol{Y}_0=(y_{10},y_{20},\cdots,y_{m0})^{\mathrm{T}}$$

则初值问题式(6.42)可改写为向量形式：

$$\begin{cases}\dfrac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}x}=\boldsymbol{f}\ (x,\boldsymbol{Y}(x))\\ \boldsymbol{Y}(a)=\boldsymbol{Y}_0\end{cases}\tag{6.43}$$

它在形式上跟单个微分方程的初值问题式(6.1)、式(6.2)完全相同,只是函数变成了向量函数.

其实,前面的讨论把函数换成向量函数也是成立的. 所以前面介绍的一切数值方法都可以推广到式(6.42)中去,只要把函数换成向量函数就行了.

例如,欧拉公式应写为

$$\boldsymbol{Y}_{n+1}=\boldsymbol{Y}_n+h\boldsymbol{f}\ (x_n,\boldsymbol{Y}_n)\tag{6.44}$$

其中 $\boldsymbol{Y}_n=(y_{1n},y_{2n},\cdots,y_{mn})^{\mathrm{T}}$

$$\approx\boldsymbol{Y}(x_n)=(y_1(x_n),y_2(x_n),\cdots,y_m(x_n))^{\mathrm{T}}$$

将式(6.44)按分量写出来,即

$$\begin{cases}y_{1,n+1}=y_{1n}+hf_1(x_n,y_{1n},y_{2n},\cdots,y_{mn})\\ y_{2,n+1}=y_{2n}+hf_2(x_n,y_{1n},y_{2n},\cdots,y_{mn})\\ \cdots\cdots\\ y_{m,n+1}=y_{mn}+hf_m(x_n,y_{1n},y_{2n},\cdots,y_{mn})\end{cases}\tag{6.45}$$

类似地,可以写出一阶常微分方程组初值问题的其他数值计算公式.

例 6.5 写出用标准四阶 R-K 方法求解两个方程的初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=f(x,y,z), & y(x_0)=y_0\\ \dfrac{\mathrm{d}z}{\mathrm{d}x}=g(x,y,z), & z(x_0)=z_0\end{cases}$$

的计算公式.

解 参照解单个微分方程的四阶 R-K 公式(6.17),我们可直接写出本题的计算公式为

$$\begin{cases} y_{n+1}=y_n+\dfrac{h}{6}(k_1+2k_2+2k_3+k_4) \\ z_{n+1}=z_n+\dfrac{h}{6}(L_1+2L_2+2L_3+L_4) \end{cases} \tag{6.46}$$

其中

$$\begin{cases} k_1=f(x_n,y_n,z_n) \\ k_2=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{h}{2}k_1,\quad z_n+\dfrac{h}{2}L_1\right) \\ k_3=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{h}{2}k_2,\quad z_n+\dfrac{h}{2}L_2\right) \\ k_4=f(x_n+h,\quad y_n+hk_3,\quad z_n+hL_3) \end{cases}$$

$$\begin{cases} L_1=g(x_n,y_n,z_n) \\ L_2=g\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{h}{2}k_1,\quad z_n+\dfrac{h}{2}L_1\right) \\ L_3=g\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{h}{2}k_2,\quad z_n+\dfrac{h}{2}L_2\right) \\ L_4=g(x_n+h,\quad y_n+hk_3,\quad z_n+hL_3) \end{cases}$$

这是一步法,利用节点 x_n 上的值 y_n,z_n,由上式顺序计算 $k_1,L_1,k_2,L_2,k_3,L_3,k_4,L_4$,然后代入式(6.46),即可求得节点 x_{n+1} 上的数值解 y_{n+1},z_{n+1}.

二、高阶微分方程的初值问题

高阶常微分方程初值问题,一般可用引进新变量,化为一阶方程组初值问题的方法求解.

下面以一般的二阶微分方程为例来说明如何构造高阶微分方程初值问题的数值计算公式.

设

$$\begin{cases} \dfrac{\mathrm{d}^2 y}{\mathrm{d}x^2}=f\left(x, y, \dfrac{\mathrm{d}y}{\mathrm{d}x}\right) \\ y(x_0)=y_0 \\ y'(x_0)=y_0' \end{cases}$$

只要引进新的变量

令

$$z=\frac{\mathrm{d}y}{\mathrm{d}x}$$

即可将上述二阶方程化为如下的一阶方程组的初值问题:

$$\begin{cases} \dfrac{\mathrm{d}z}{\mathrm{d}x}=f(x, y, z), & z(x_0)=y_0' \\ \dfrac{\mathrm{d}y}{\mathrm{d}x}=z, & y(x_0)=y_0 \end{cases}$$

这时,其四阶龙格-库塔格式具有形式:

$$\begin{cases} z_{n+1}=z_n+\dfrac{h}{6}(k_1+2k_2+2k_3+k_4) \\ y_{n+1}=y_n+\dfrac{h}{6}(L_1+2L_2+2L_3+L_4) \end{cases}$$

其中

$$\begin{cases} k_1=f(x_n, y_n, z_n) \\ k_2=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{h}{2}L_1,\quad z_n+\dfrac{h}{2}k_1\right) \\ k_3=f\left(x_n+\dfrac{h}{2},\quad y_n+\dfrac{h}{2}L_2,\quad z_n+\dfrac{h}{2}k_2\right) \\ k_4=f(x_n+h,\quad y_n+hL_3,\quad z_n+hk_3) \end{cases}$$

$$
\begin{cases}
L_1 = z_n \\
L_2 = z_n + \dfrac{h}{2}k_1 \\
L_3 = z_n + \dfrac{h}{2}k_2 \\
L_4 = z_n + hk_3
\end{cases}
$$

以上二式中可以消去 L_1, L_2, L_3 和 L_4，则龙格-库塔格式可进一步简化为

$$
\begin{cases}
z_{n+1} = z_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
y_{n+1} = y_n + hz_n + \dfrac{h^2}{6}(k_1 + k_2 + k_3)
\end{cases}
$$

这里

$$
\begin{cases}
k_1 = f(x_n, y_n, z_n) \\
k_2 = f\left(x_n + \dfrac{h}{2},\quad y_n + \dfrac{h}{2}z_n,\quad z_n + \dfrac{h}{2}k_1\right) \\
k_3 = f\left(x_n + \dfrac{h}{2},\quad y_n + \dfrac{h}{2}z_n + \dfrac{h^2}{4}k_1,\quad z_n + \dfrac{h}{2}k_2\right) \\
k_4 = f\left(x_n + h,\quad y_n + hz_n + \dfrac{h^2}{2}k_2,\quad z_n + hk_3\right)
\end{cases}
$$

即得到了进一步简化了的只含有 $k_i(i=1,2,3,4)$ 的龙格-库塔公式.

学习指导

一、基本要求与重点

1. **了解常微分方程初值问题数值解法的一些基本概念，如单**

步法和多步法，显式和隐式，方法的阶数，整体截断误差和局部截断误差的区别和关系等．

2．掌握一阶常微分方程初值问题的一些常用的数值计算方法，例如欧拉（Euler）方法、改进的欧拉方法、龙格-库塔（Runge-Kutta）方法、阿达姆斯（Admas）方法等，要注意各方法的特点及有关的理论分析．

3．掌握构造常微分方程数值的数值积分的构造方法和泰勒展开的构造方法的基本思想，并能具体应用它们导出一些常用的数值计算公式及估计截断误差．

4．熟练掌握龙格-库塔（R-K）方法的基本思想，公式的推导，R-K公式中系数的确定，特别是能应用“标准四阶R-K公式”解题．

5．掌握数值方法的收敛性和稳定性的概念，并能确定给定方法的绝对稳定性区域．

二、例题分析与解答

例1　对常微分方程初值问题

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x}=-y \\ y(0)=1 \end{cases} \qquad (0\leqslant x\leqslant 1)$$

取步长 $h=0.1$，分别用欧拉方法、改进的欧拉方法及标准的四阶R-K方法作数值计算，并将计算结果与方程的精确解 $y(x)=\mathrm{e}^{-x}$ 进行比较．

解　由欧拉公式(6.3)改进的欧拉公式(6.5)及标准的四阶R-K公式(6.17)，可得三种方法的具体计算公式分别如下：

(1) 欧拉方法：

$$y_{n+1}=y_n+0.1(-y_n)=0.9y_n$$

(2) 改进的欧拉方法：

$$\begin{cases} \bar{y}_{n+1} = y_n + 0.1(-y_n) = 0.9y_n \\ y_{n+1} = y_n + \frac{0.1}{2}[(-y_n) + (-\bar{y}_{n+1})] = 0.905y_n \end{cases}$$

（3）**标准的四阶 R-K 方法：**

$$\begin{cases} y_{n+1} = y_n + \frac{0.1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 0.9048375y_n \\ k_1 = -y_n \\ k_2 = -(y_n + 0.05k_1) = -0.95y_n \\ k_3 = -(y_n + 0.05k_2) = -0.9525y_n \\ k_4 = -(y_n + 0.1k_3) = -0.90475y_n \end{cases}$$

计算结果与精确解的比较如表(例)6-1 所示．

表(例)6-1

	欧拉方法	**改进的欧拉方法**	**标准四阶 R-K 方法**	**精确解法**
x_n	y_n	y_n	y_n	$y(x_n)$
0.1	0.900000	0.905000	0.904837	0.904837
0.2	0.810000	0.819025	0.818730	0.818730
0.3	0.729000	0.741218	0.740818	0.740818
0.4	0.656100	0.670802	0.670320	0.670320
0.5	0.590490	0.607076	0.606530	0.606530
0.6	0.531441	0.549404	0.548812	0.548811
0.7	0.478297	0.497210	0.496585	0.496585
0.8	0.430467	0.449975	0.449329	0.449328
0.9	0.387421	0.407228	0.406569	0.406569
1.0	0.348679	0.368541	0.367879	0.367879

可见，标准四阶 R-K 方法计算的精确度最高．

例 2 利用欧拉方法计算积分

$$\int_0^x e^{t^2}\,dt$$

在点 $x=0.5,1,1.5,2$ 处的数值解．

解 令 $y(x)=\int_0^x e^{t^2}\,dt$，则有 $\frac{dy}{dx}=e^{x^2}$，$y(0)=0$．对此一阶常微分方程初值问题，取步长 $h=0.5$，由欧拉公式(6.3)得其具体计算公式为

$$\begin{cases} y_{n+1}=y_n+0.5e^{x_n^2} \\ y(0)=y_0=0 \end{cases}$$

则用欧拉方法计算积分 $\int_0^x e^{t^2}\,dt$ 的数值解如表(例)6-2 所示．

表(例)6-2

x_n	0.5	1.0	1.5	2.0
y_n	0.500000	1.14201	2.50115	7.24502

例 3 用欧拉方法解初值问题

$$\begin{cases} \frac{dy}{dx}=ax+b \\ y(0)=0 \end{cases}$$

证明：其整体截断误差为

$$e_n=y(x_n)-y_n=\frac{1}{2}anh^2,$$

这里，$x_n=nh$，y_n 是欧拉方法的数值解．

证 显然,该初值问题的精确解为

$$y(x)=\frac{1}{2}ax^2+bx\text{,则有 } y(x_n)=\frac{1}{2}ax_n^2+bx_n.$$

由欧拉公式(6.3),有

$$y_n=y_{n-1}+hf(x_{n-1},y_{n-1})\overset{\text{记}}{=\!=\!=}y_{n-1}+hf_{n-1}$$

$$(n=1,2,\cdots)$$

今将已知 $f(x,y)=ax+b$,代入上述递推式,有

$$\begin{aligned}
y_n &=y_{n-1}+hf_{n-1}=(y_{n-2}+hf_{n-2})+hf_{n-1}\\
&=(y_{n-3}+hf_{n-3})+hf_{n-2}+hf_{n-1}\\
&\cdots\\
&=y_0+h\sum_{i=0}^{n-1}f_i=0+h\sum_{i=0}^{n-1}(ax_i+b)\\
&=nbh+ha\sum_{i=0}^{n-1}x_i=nbh+ah(x_0+x_1+\cdots+x_{n-1})\\
&=nbh+ah(0+h+2h+\cdots+nh)\\
&=nbh+ah^2\cdot\frac{1+(n-1)}{2}\cdot(n-1)\\
&=bx_n+\frac{1}{2}ax_n^2-\frac{1}{2}ahx_n.
\end{aligned}$$

而 $$y(x_n)=\frac{1}{2}ax_n^2+bx_n$$

所以 $$\begin{aligned}
e_n &=y(x_n)-y_n\\
&=\frac{1}{2}ax_n^2+bx_n-\left(bx_n+\frac{1}{2}ax_n^2-\frac{1}{2}ahx_n\right)
\end{aligned}$$

$$=\frac{1}{2}ahx_n=\frac{1}{2}ah(nh)=\frac{1}{2}anh^2.$$

例 4　用泰勒展开方法求解初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=x^2+y^2\\ y(x_0)=y_0\end{cases}$$

解　泰勒展开的方法也是微分方程离散化的一个重要方法．这个方法的具体做法是，假设初值问题式(6.1)～**式**(6.2)**的精确解** $y(x)$ **在** $a\leqslant x\leqslant b$ **上存在** $p+1$ **阶连续导数，那么，**$y(x_{n+1})$ **在** $x=x_n$ **处作泰勒展开，有**

$$\begin{aligned}y(x_{n+1})&=y(x_n+h)\\&=y(x_n)+hy'(x_n)+\frac{h^2}{2!}y''(x_n)+\frac{h^2}{3!}y'''(x_n)\\&\quad+\cdots+\frac{h^p}{p!}y^{(p)}(x_n)+R_p(x)\end{aligned}\tag{1}$$

其中截断误差为

$$R_p(x)=\frac{h^{p+1}}{(p+1)!}y^{(p+1)}(\xi)=O(h^{p+1})\quad(x_n<\xi<x_{n+1})$$

在式(1)**中略去截断误差，并以近似值** $y_n^{(k)}$ **代替** $y^{(k)}(x_n)$**，则得**

$$y_{n+1}=y_n+hy_n'+\frac{h^2}{2!}y_n''+\cdots+\frac{h^p}{p!}y_n^{(p)}\tag{2}$$

用公式(2)**解初值问题式**(6.1)～**式**(6.2)**的数值方法称为泰勒展开法，由于它的截断误差是** $O(h^{p+1})$**，故称为** p **阶方法．**

对于本题，用泰勒展开法的关键是先求出 y **的各阶导数，再代入泰勒展开式**(1)**，由已知** $y'=x^2+y^2$**，直接求其各阶导数，得**

$$y''=2x+2yy'=2x+2y(x^2+y^2)$$

$$y'''=2+2yy''+2(y')^2$$

$$=2+4xy+2(x^2+y^2)(x^2+3y^2)$$

$$y^{(4)}=2yy'''+2y'y''+4y'y''=2yy'''+6y'y''$$

$$=4y+4x(3x^2+5y^2)+8y(x^2+y^2)(2x^2+3y^2)$$

$$\vdots$$

当然,可以继续求导下去,但随着求导次数增大,工作量越来越大,如果要求得到公式的局部截断误差为 $O(h^4)$,**则把上列求得的各阶导数代入泰勒展开式**(1)**中,并设** $y(x_n)=y_n$,**我们就构造了一个求解微分方程**$\frac{dy}{dx}=x^2+y^2$ **的数值公式:**

$$y_{n+1}=y_n+hf_n+\frac{1}{2}h^2f'_n+\frac{1}{6}h^2f''_n$$

其中,$f_n^{(k)}$ **表示** $f(x,y)$**对** x **的** k **阶偏导数在** $x=x_n$ **点的值. 由于这个公式的局部截断误差是** $O(h^4)$,**故称为三阶方法.**

从理论上讲,只要方程的解 $y(x)$**充分光滑,用泰勒展开方法可以构造任意有限阶的数值计算公式,但由于求复合函数的高阶导数往往是很繁琐的,因此,泰勒展开法一般只用于求"表头",即开头几点的数值解,如** y_1,y_2,y_3,y_4 **等.**

例 5　取步长 $h=0.4$,**写出用标准四阶** R-K **方法求解初值问题**

$$\begin{cases}\dfrac{dy}{dx}=x\sin(x+y)\\ y(1)=0\quad(1\leqslant x\leqslant 9)\end{cases}$$

的计算公式.

解　$x_n=x_0+nh=1+n\times0.4=1+0.4n$, $y_0=0$,**取** $n=0,1,2,\cdots,20$,**其标准四阶** R-K **方法的计算公式为**

$$\begin{cases} y_{n+1}=y_n+\frac{0.4}{6}(k_1+2k_2+2k_3+k_4) \\ k_1=(1+0.4n)\sin(1+0.4n+y_n) \\ k_2=(1.2+0.4n)\sin(1.2+0.4n+y_n+0.2k_1) \\ k_3=(1.2+0.4n)\sin(1.2+0.4n+y_n+0.2k_2) \\ k_4=(1.4+0.4n)\sin(1.4+0.4n+y_n+0.4k_3) \end{cases}$$

$$(n=0,1,2,\cdots,19)$$

例 6　用数值积分方法导出求解初值问题

$$\begin{cases} \frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y) \\ y(x_0)=y_0 \end{cases}$$

的中矩形公式：

$$y_{n+1}=y_{n-1}+2hf(x_n,y_n)$$

并估计其局部截断误差.

解　设 $h=x_n-x_{n-1}=x_{n+1}-x_n$，对 $\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$ 在区间 $[x_{n-1},x_{n+1}]$ 上积分，得

$$y(x_{n+1})=y(x_{n-1})+\int_{x_{n-1}}^{x_{n+1}} f(x,y(x))\mathrm{d}x$$

$$=y(x_{n+1})+\int_{x_{n-1}}^{x_{n+1}} y'(x)\mathrm{d}x$$

$$=y(x_{n-1})+\int_{x_{n-1}}^{x_{n+1}} \Big[y'(x_n)+(x-x_n)y''(x_n)$$

$$+\frac{(x-x_n)^2}{2}y'''(\xi)\Big]\mathrm{d}x=y(x_{n-1})$$

$$+y'(x_n)x\Big|_{x_{n-1}}^{x_{n+1}}+\frac{(x-x_n)^2}{2}y(x_n)\Big|_{x_{n-1}}^{x_{n+1}}$$

$$+y'''(\eta)\int_{x_{n-1}}^{x_{n+1}}\frac{(x-x_n)^2}{2}\mathrm{d}x$$

分别用 y_{n-1},y_n,y_{n+1} 代上式中的 $y(x_{n-1})$，$y(x_n),y(x_{n+1})$，得

$$y_{n+1}=y_{n-1}+2hy'(x_n)+\frac{h^3}{3}y'''(\eta)$$

则得到求解初值问题的中矩形公式：

$$y_{n+1}=y_{n-1}+2hf(x_n,y_n)$$

其局部截断误差为

$$R=\frac{1}{3}h^3y'''(\eta)=O(h^3),\quad \eta\in(x_{n-1},x_{n+1})$$

所以，中矩形公式具有二阶精度．

例 7　用泰勒展开的构造方法构造一个以 $x_{n-2},x_{n-1},x_n,x_{n+1}$ 为节点的线性三步公式：

$$y_{n+1}=a_0y_n+a_1y_{n-1}+a_2y_{n-2}+h(\beta_{-1}f_{n+1}$$

$$+\beta_0f_n+\beta_1f_{n-1})\tag{3}$$

解　利用泰勒展开的方法构造线性多步法的计算公式的基本思想类同于 R-K 方法的推导，它是先将线性多步法公式(6.25)的右端均在 $x=x_n$ 处作泰勒展开，然后按 h 的幂次作升幂排列重新整理得

$$y_{n+1}=(\sum_{k=0}^{r}\alpha_k)y_n+\sum_{j=1}^{p}\frac{h^j}{j!}[\sum_{k=1}^{r}(-k)^j\alpha_k$$

$$+j\sum_{k=-1}^{r}(-k)^{j-1}\beta_k]y_n^{(j)}+\frac{h^{p+1}}{(p+1)!}[\sum_{k=1}^{r}(-k)^{p+1}\alpha_k$$

$$+(p+1)\sum_{k=-1}^{r}(-k)^p\beta_k]y_n^{p+1}+\cdots\tag{4}$$

于是，欲使公式(6.25)成为 p 阶的，即局部截断误差为

$O(h^{p+1})$，只要对展开式(4)与 $y(x_{n+1})$ 在 x_n 处的泰勒展开式

$$y(x_{n+1})=\sum_{j=0}^{p}\frac{h^j}{j!}y_n^{(j)}+\frac{h^{p+1}}{(p+1)!}y_n^{(p+1)}+\cdots \tag{5}$$

相比较，能符合到 h^p 项，为此令

$$\begin{cases}\sum_{k=0}^{r}\alpha_k=1\\ \sum_{k=1}^{r}(-k)^j\alpha_k+j\sum_{k=1}^{r}(-k)^{j-1}\beta_k=1\end{cases} \tag{6}$$

$$(j=1,2,\cdots,p)$$

解出系数 α_k，β_k，代入式(6.26)，即得 p 阶的线性多步法公式．式(5)与式(4)相减，可得方法的局部截断误差为

$$y(x_{n+1})-y_{n+1}=\frac{h^{p+1}}{(p+1)!}\Big[1-\sum_{k=1}^{r}(-k)^{p+1}\alpha_k$$

$$-(p+1)\sum_{k=-1}^{r}(-k)^p\beta_k\Big]y_n^{(p+1)}+\cdots \tag{7}$$

对本题，显然，关键只要确定公式中的系数 α_0，α_1，α_2，β_{-1}，β_0，β_1，设

$$y_i=y(x_i)\ (i=n-2,n-1,n)$$

$$f(x_i,y_i)=y'(x_i)\ (i=n-1,n,n+1)$$

然后将函数 $y(x_{n-2})$，$y(x_{n-1})$，$y'(x_{n-1})$，$y'(x_{n+1})$ 在点 $x=x_n$ 处作泰勒展开后，代入式(3)，再按 h 的升幂排列整理得

$$y_{n+1}=(\alpha_0+\alpha_1+\alpha_2)y(x_n)$$

$$+(-\alpha_1-2\alpha_2+\beta_{-1}+\beta_0+\beta_0)y'(x_n)h$$

$$+\left(\frac{1}{2}\alpha_1+2\alpha_2+\beta_{-1}-\beta_1\right)y''(x_n)h^2$$

$$+\left(-\frac{1}{6}\alpha_1-\frac{4}{3}\alpha_2+\frac{1}{2}\beta_{-1}+\frac{1}{2}\beta_1\right)y'''(x_n)h^3$$

$$+\left(\frac{1}{24}\alpha_1+\frac{2}{3}\alpha_2+\frac{1}{6}\beta_{-1}-\frac{1}{6}\beta_1\right)y^{(4)}(x_n)h^4$$

$$+\left(-\frac{1}{120}\alpha_1-\frac{32}{120}\alpha_2+\frac{1}{24}\beta_{-1}+\frac{1}{24}\beta_1\right)y^{(5)}(x_n)h^5$$

$$+\cdots \tag{8}$$

另一方面，将 $y(x_{n+1})=y(x_n+h)$ 在 x_n 处作泰勒展开，有

$$y(x_{n+1})=y(x_n)+y'(x_n)h+\frac{1}{2!}y''(x_n)h^2$$

$$+\frac{1}{3!}y'''(x_n)h^3+\frac{1}{4!}y^{(4)}(x_n)h^4$$

$$+\frac{1}{5!}y^{(5)}(x_n)h^5+\cdots \tag{9}$$

显然，要使公式成为五阶，也即局部截断误差为 $y(x_{n+1})-y_{n+1}=O(h^6)$，则要使公式(8)与式(9)中的前六项关于 h 的同次幂系数完全相同，由并 $y(x)$ 的任意性可得六个方程：

$$\begin{cases}h^0:\ \alpha_0+\alpha_1+\alpha_2=1\\ h^1:-\alpha_1-2\alpha_2+\beta_{-1}+\beta_0+\beta_1=1\\ h^2:\frac{1}{2}\alpha_1+2\alpha_2+\beta_{-1}-\beta_1=\frac{1}{2}\\ h^3:-\frac{1}{6}\alpha_1-\frac{4}{3}\alpha_2+\frac{1}{2}\beta_{-1}+\frac{1}{2}\beta_1=\frac{1}{6}\\ h^4:\frac{1}{24}\alpha_1+\frac{2}{3}\alpha_2+\frac{1}{6}\beta_{-1}-\frac{1}{6}\beta_1=\frac{1}{24}\\ h^5:-\frac{1}{120}\alpha_1-\frac{32}{120}\alpha_2+\frac{1}{24}\beta_{-1}+\frac{1}{24}\beta_1=\frac{1}{120}\\ \cdots\end{cases} \tag{10}$$

从式(10)的前六个方程中解出 6 个未知元 $\alpha_0,\alpha_1,\alpha_2,\beta_{-1},\beta_0,\beta_1$，再代入式(3)中，即得到一个五阶的数值计算公式．当然，我们

也可以只要求前面几个方程成立，譬如说，我们只要求前面四个方程成立，则从前面四个方程中解 6 个未知元，这样，就有两个自由参数，如令 $\alpha_0=0$，$\alpha_2=0$，这时，从前面四个方程中可解得 $\alpha_1=1$，$\beta_{-1}=\frac{1}{3}$，$\beta_0=\frac{4}{3}$，$\beta_1=\frac{1}{3}$，代入式(3)，得到计算公式：

$$y_{n+1}=y_{n-1}+\frac{h}{3}(f_{n+1}+4f_n+f_{n-1}) \tag{11}$$

称为辛甫生公式，此公式也可用数值积分得到．

从上面的推导过程知此公式的局部截断误差是 $O(h^4)$，但这时，式(10)中第五个方程恰巧也满足，所以阶数可再提高一阶，是一个 4 阶公式，但此公式的稳定性较差．哈明用 α_1 的不同数值进行试验发现，若令 $\alpha_1=0$，并要求式(10)中的前 5 个方程成立，则可得

$$\alpha_0=\frac{9}{8},\ \alpha_2=-\frac{1}{8},\ \beta_{-1}=\frac{3}{8},\ \beta_0=\frac{6}{8},\beta_1=-\frac{3}{8}$$

由此得稳定性较好的哈明公式：

$$y_{n+1}=\frac{1}{8}[9y_n-y_{n-2}+3h(f_{n+1}+2f_n-f_{n-1})] \tag{12}$$

此式是隐式三步法，这个公式不能用数值积分方法推得，可证得其局部截断误差为

$$R=y(x_{n+1})-y_{n+1}=\frac{-1}{40}h^5y^{(5)}(\xi).$$

例 8 对初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=-y\\ y(0)=1\end{cases}$$

证明：用梯形法求得的数值解为 $y_n=\left(\frac{2-h}{2+h}\right)^n$，并证明当步长 $h\to$

0 时，y_n 收敛于初值问题的精确解 $y(x)=e^{-x}$.

证 梯形法的计算公式为

$$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},y_{n+1})]$$

本题中，$f(x,y)=-y$，代入梯形公式中，得计算式：

$$y_{n+1}=y_n+\frac{h}{2}(-y_n-y_{n+1}),$$

则有
$$y_n=y_{n-1}+\frac{h}{2}(-y_{n-1}-y_n)$$

经整理得
$$y_n=\frac{2-h}{2+h}y_{n-1}$$

从而有

$$y_n=\frac{2-h}{2+h}y_{n-1}=\left(\frac{2-h}{2+h}\right)^2 y_{n-2}=\cdots=\left(\frac{2-h}{2+h}\right)^n\cdot y_0$$

由已知 $y_0=1$，故证明了用梯形法求得的数值解为

$$y_n=\left(\frac{2-h}{2+h}\right)^n.$$

以下证明当 $h\to 0$ 时，有 $y_n\to e^{-x}$：此处，$x=nh$，计算极限

$$\lim_{h\to 0}\left(\frac{2-h}{2+h}\right)^n=\lim_{h\to 0}\left(1-\frac{2h}{2+h}\right)^{\frac{x}{h}}=\lim_{h\to 0}\frac{1}{\left(1-\frac{2h}{2+h}\right)^{-\frac{x}{h}}}$$

$$=\lim_{h\to 0}\frac{1}{\left[\left(1-\frac{2h}{2+h}\right)^{-\frac{2+h}{2h}}\cdot\left(1-\frac{2h}{2+h}\right)^{\frac{1}{2}}\right]^x}=\frac{1}{e^x}=e^{-x}.$$

这说明，对于本题的初值问题，梯形法是收敛的.

例 9 讨论梯形法的绝对稳定性区域.

解 由定义 6.4 知,把梯形法的计算公式用于试验方程$\dfrac{dy}{dx}=\lambda y$(其中λ为常数,当λ是复数时,$R_e(\lambda)<0$)得其计算公式为

$$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},y_{n+1})]$$

$$=y_n+\frac{h}{2}[-\lambda y_n-\lambda y_{n+1}]$$

若y_n的实际计算值为$\tilde{y}_n$,则由误差$\delta_n=y_n-\tilde{y}_n$引起y_{n+1}的误差为$\delta_{n+1}=\left(\dfrac{2+\lambda h}{2-\lambda h}\right)\delta_n$.

由此可见,要使y_n的误差δ_n逐步削弱,$\mu=\lambda h$应满足$\left|\dfrac{2+\mu}{2-\mu}\right|<1$,显然,对于任意的步长$h>0$都满足,也即对步长$h$不需作任何限制,梯形法对$0<h<+\infty$都是绝对稳定的,并称它为 *A*-稳定.

例 10 证明:对初值问题

$$\begin{cases}\dfrac{dy}{dx}=f(x,y)\\ y(a)=y_0 \quad (a\leqslant x\leqslant b)\end{cases}$$

当方程的右端函数$f(x,y)$在其定义区域上关于y满足李普希兹条件时,标准 R-K 方法是收敛的.

证 由标准 R-K 方法的计算公式(6.17)和定理 6.1 知:当假设$f(x,y)$在区域$a\leqslant x\leqslant b,-\infty<y<+\infty$上关于$y$满足李普希兹条件,即

$$|f(x,y_1)-f(x,y_2)|\leqslant L(y_1-y_2|$$

很容易验证标准四阶 R-K 方法的增量函数$\phi(x,y,h)$关于y也满足李普希兹条件,从而标准四阶 R-K 方法收敛.

显然,R-K 方法的增量函数

$$\phi(x,y,h)=\frac{1}{6}[k_1(x,y,h)+2k_2(x,y,h)$$

$$+2k_3(x,y,h)+k_4(x,y,h)]$$

其中 $k_1(x,y,h)=f(x,y)$

$$k_2(x,y,h)=f\left(x+\frac{h}{2},\ y+\frac{hk_1(x,y,h)}{2}\right)$$

$$k_3(x,y,h)=f\left(x+\frac{h}{2},\ y+\frac{hk_2(x,y,h)}{2}\right)$$

$$k_4(x,y,h)=f(x+h,y+hk_3(x,y,h))$$

由于 $f(x,y)$ **关于** y **满足李普希兹条件，从而**

$$|k_1(x,y,h)-k_1(x,y^*,h)|\leqslant L|y-y^*|$$

$$|k_2(x,y,h)-k_2(x,y^*,h)|$$

$$=\left|f\left(x+\frac{h}{2},\ y+\frac{hk_1(x,y,h)}{2}\right)\right.$$

$$\left.-f\left(x+\frac{h}{2},y^*+\frac{hk_1(x,y^*,h)}{2}\right)\right|$$

$$\leqslant L\left|y+\frac{hk_1(x,y,h)}{2}-\left(y^*+\frac{hk_1(x,y^*,h)}{2}\right)\right|$$

$$\leqslant L\left(|y-y^*|+\frac{1}{2}hL|y-y^*|\right)$$

$$=L\left(1+\frac{1}{2}hL\right)|y-y^*|$$

同理，可得以下不等式：

$$|k_3(x,y,h)-k_3(x,y^*,h)|$$

$$\leqslant L\left[\left(1+\frac{1}{2}hL\right)+\frac{1}{4}(hL)^2\right]|y-y^*|,$$

$$|k_4(x,y,h)-k_4(x,y^*,h)|$$

$$\leqslant L\left[1+hL+\frac{1}{2}(hL)^2+\frac{1}{4}(hL)^3\right]|y-y^*|$$

于是，作为 $k_i(i=1,2,3,4)$ **线性组合的** $\phi(x,y,h)$ **在** $a\leqslant x\leqslant b$，$-\infty<y<+\infty$，$0\leqslant h\leqslant h_0$ **上有**

$$|\phi(x,y,h)-\phi(x,y^*,h)|$$

$$\leqslant L\left[1+\frac{1}{2}h_0L+\frac{1}{6}(h_0L)^2+\frac{1}{24}(h_0L)^3\right]$$

$$\cdot|y-y^*|=\bar{L}|y-y^*|$$

即 $\phi(x,y,h)$ **关于** y **满足李普希兹条件，其李普希兹常数为**

$$\bar{L}=L\left[1+\frac{1}{2}h_0L+\frac{1}{6}(h_0L)^2+\frac{1}{24}(h_0L)^3\right]$$

所以，标准四阶 R-K **方法收敛**.

例 11　分别用四阶阿达姆斯显式和隐式公式求初值问题

$$\begin{cases}\dfrac{dy}{dx}=-y+x+1\\ y(0)=1,\quad(0\leqslant x\leqslant 1)\end{cases}$$

的数值解，取步长 $h=0.1$.

解　本题的精确解为 $y(x)=e^{-x}+x$，**根据题意，有**

$$x_n=nh=0.1n,\ f_n=-y_n+x_n+1$$

由四阶阿达姆斯显式公式(6.29)，**有**

$$y_{n+1}=y_n+\frac{h}{24}(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3})$$

$$=\frac{1}{24}[18.5y_n+5.9y_{n-1}-3.7y_{n-2}+0.9y_{n-3}$$

$$+0.24n+2.52], \quad (n=3,4,\cdots,9)$$

由四阶阿达姆斯隐式公式(6.30),有

$$y_{n+1}=y_n+\frac{h}{24}(9f_{n+1}+19f_n-5f_{n-1}+f_{n-2})$$

$$=\frac{1}{24}[-0.9y_{n+1}+22.1y_n+0.5y_{n-1}-0.1y_{n-2}$$

$$+0.24n+2.52]$$

由上式可解出

$$y_{n+1}=\frac{1}{24.9}[22.1y_n+0.5y_{n-1}-0.1y_{n-2}$$

$$+0.24n+2.52], \quad (n=2,3,\cdots,9)$$

利用精确解 $y(x)=e^{-x}+x$,求出起步值(一般可用四阶 R-K 公式计算起步值)后,按上面的公式计算,其结果如表(例)6-3 所示.

表(例)6-3

x_n	阿达姆斯显式法		阿达姆斯隐式法	
	y_n	$\lvert y(x_n)-y_n\rvert$	y_n	$\lvert y(x_n)-y_n\rvert$
0.3	初　值		1.04081800	2.146×10^{-7}
0.4	1.07032292	2.874×10^{-6}	1.07031966	3.846×10^{-7}
0.5	1.10653547	4.816×10^{-6}	1.10653013	5.213×10^{-7}
0.6	1.14881840	6.772×10^{-6}	1.14881100	6.285×10^{-7}
0.7	1.19659339	8.090×10^{-6}	1.19658459	7.106×10^{-7}
0.8	1.24933815	9.192×10^{-6}	1.24932819	7.714×10^{-7}
0.9	1.30657961	9.954×10^{-6}	1.30656884	8.141×10^{-7}
1.0	1.36788995	1.052×10^{-5}	1.36787859	8.418×10^{-7}

从表(例)6-3 中可以看出,四阶阿达姆斯隐式法比显式法的精度高,一般地,同阶的隐式法比显式法精确,而且数值稳定性也

好．但在隐式公式中，通常很难解出 y_{n+1}，需要用迭代法求解，这样，又增加了计算量．因此，实际计算时，很少单独使用显式公式或隐式公式，而是将它们联合使用：先用显式公式求出 $y(x_{n+1})$ 的预测值，记作 $\bar{y}_{n+1}$，再用隐式公式对预测值进行校正，求出 $y(x_{n+1})$ 的近似值，例如 $r=3$ 时的公式(6.30)．

习　题　六

1．用欧拉方法求解初值问题

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x}=-y+x+1 \\ y(0)=1, \quad (0\leqslant x\leqslant 1, h=0.1). \end{cases}$$

2．取步长 $h=0.2$，用标准四阶 R-K 方法求解初值问题

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x}=x+y \\ y(0)=1 \end{cases}$$

$$(0\leqslant x\leqslant 1)$$

并将计算结果与精确解比较.

3．验证公式

$$y_{n+1}=y_n+h\psi(x_n, y_n, h)$$

是二阶 R-K 方法，这里

$$\psi(x,y,h)=\frac{1}{2}[f(x,y)+f(x+h, y+hf(x,y))]$$

若 L_0 是 f 关于 y 的李普希兹常数，验证

$$L=L_0\left(1+\frac{h_0}{2}L_0\right)$$

是 ψ 关于 y 的李普希兹常数.

4．对初值问题 $y'=-10y, y(a)=y_0$，由第 3 题给出的推算公式，讨论绝对稳定性对步长 h 的限制.

5．证明：对任意参数 t，下列 R-K 公式

$$\begin{cases} y_{n+1}=y_n+\frac{1}{2}(k_2+k_3) \\ k_1=hf(x_n, y_n) \\ k_2=hf(x_n+th, y_n+tk_1) \\ k_3=hf(x_n+(1-t)h, y_n+(1-t)k_1) \end{cases}$$

的局部截断误差为 $O(h^3)$.

6．求系数 a, b, c 和 d，使数值计算公式

$$y_{n+1}=ay_{n-1}+h(by'_{n+1}+cy'_n+dy'_{n-1})$$

有

$$y(x_{n+1})-y_{n+1}=O(h^5).$$

附录Ⅰ　部分数值方法的计算实例

随着科学技术的发展和计算机的广泛使用，数值计算几乎在所有的科技领域中迅速地发展，人们愈来愈认识到科学计算是科学研究的继实验方法、理论方法之后的第三种方法. 本教材介绍的各种数值计算方法都有相应的标准程序，有了数值计算的理论基础，就可以在实践中更好地应用和修改程序，编制适用的程序软件，以灵活地适应实际问题的需要，下面给出几个数值方法的应用实例，其程序均采用FORTRAN77语言编写，并全部在PC-486微机上用NDP软件进行调试通过.

一、龙贝格求积算法

1. 方法简介

用龙贝格方法计算定积分 $I=\int_a^b f(x)\mathrm{d}x$ 的近似值的步骤如下：

(1) 准备初值

先用梯形公式计算：

$$T_1=\frac{b-a}{2}[f(a)+f(b)]$$

(2) 按变步长梯形法则计算积分近似值

将区间$[a,b]$逐次折半，令区间长度

$$h=\frac{b-a}{2^i},\quad (i=1,2,\cdots)$$

计算

$$T_{2n}=\frac{1}{2}T_n+h\sum_{k=1}^{n-1}\underset{\sim\sim\sim}{f[a+(2k-1)h]},\left(\text{其中 } n=2^{i-1},h=\frac{b-a}{2n}\right)$$

↑——新分点上的函数值

(3) 按外推公式求积分加速值

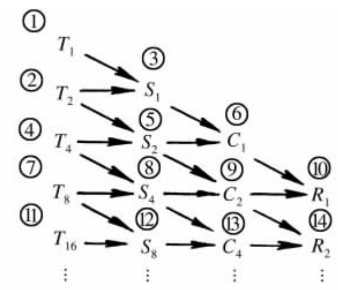
抛物线求积公式：$S_n=T_{2n}+(T_{2n}-T_n)/3$；

柯特斯求积公式：$C_n=S_{2n}+(S_{2n}-S_n)/15$；

龙贝格求积公式：$R_n=C_{2n}+(C_{2n}-C_n)/63$.

龙贝格求积算法的计算过程可以构造附表 1 所示的龙贝格表.

附表 1　　龙贝格表



(4) 精度控制

直到相邻两次的积分近似值 R_{2n} 和 R_n 满足如下关系：

当 $|R_{2n}|\leqslant 1$ 时，满足 $|R_{2n}-R_n|<\varepsilon$

当 $|R_{2n}|>1$ 时，满足 $\left|\frac{R_{2n}-R_n}{R_{2n}}\right|<\varepsilon$

其中，ε 为允许的误差限.

2. 程序及例

(1) 程序要点

ROMBERG 算法产生一个三角数组，这个数组可用记号表示

如下：

$$
\begin{array}{llllll}
R(1,1) & & & & & \\
R(2,1) & R(2,2) & & & & \\
R(3,1) & R(3,2) & R(3,3) & & & \\
R(4,1) & R(4,2) & R(4,3) & R(4,4) & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
R(N,1) & R(N,2) & R(N,3) & R(N,4) & \cdots & R(N,N)
\end{array}
$$

其中的每一个数都是对定积分 $\int_a^b f(x)\mathrm{d}x$ 的数值估计，并有

$$\lim_{N\to\infty} R(N,1)=\lim_{N\to\infty} R(N,N)$$

第 1 列 $R(i,1)$ 是 2^i 个相等子区间运用梯形法则的结果：

$$R(1,1)=\frac{1}{2}(b-a)(f(a)+f(b))$$

$$R(i,1)=\frac{1}{2}R(i-1,1)+h\sum_{k=1}^{2^{i-1}-1} f(a+(2k-1)h)$$

$$(h=(b-a)/2^i,\quad i>1)$$

第 2 列以及后面的各列是由外推公式产生：

$$R(i,j)=R(i,j-1)+\frac{1}{4^{j-1}-1}(R(i,j-1)-R(i-1,j-1))$$

本程序是逐行产生 ROMBERG 数组.

(2) 子程序 ROMBERG(F,A,B,N,R,ER)变量说明

F——被积函数，以函数子程序给出；

A,B——积分的下上限；

R——ROMBERG 数组；

N——计算步数；

ER——计算误差.

(3) **运用本程序对定积分 $\int_0^{\pi} \sin x \mathrm{d}x$ 进行计算实例**

(4) **程序**

```
C
C
C  THIS PROGRAM IS ROMBERG METHOD. IT CONTINUES ONE SUB-
C  ROUTINE AND ONE FUNCTION AND ONE MAIN PROGRAM.
C      F——FUNCTION;
C      A,B——INTERNAL NODE;
C      N——CALCULATE STEPS;
C      R——VALUE;
C      ER——CONTRALED ERROR;
C
C
C
      SUBROUTINE ROMBRG(F,A,B,N,R,ER)
      DIMENSION R(N,N)
      H=B-A
      R(1,1)=0.5*H*(F(A)+F(B))
      OPEN(10,FILE='OUT1.DAT')
      WRITE(10,5)R(1,1)
      L=1
      DO 4 I=2,N
      H=0.5*H
      L=L+L
      SUM=0.
      DO 2 K=1,L-1,2
      SUM=SUM+F(A+REAL(K)*H)
2     CONTINUE
      R(I,1)=0.5*R(I-1,1)+H*SUM
      M=1
      DO 3 J=2,I
      M=4*M
      R(I,J)=R(I,J-1)+(R(I,J-1)-R(I-1,J-1))/REAL(M-1)
```

```
3       CONTINUE
        WRITE(10,5)(R(I,J),J=1,I)
        IF(ABS(R(I,J)-R(I,J-1)).LE.ER)RETURN
4       CONTINUE
        CLOSE(10)
5       FORMAT(5X,5E14.7)
        RETURN
        END

        FUNCTION F(X)
        F=SIN(X)
        RETURN
        END

        EXTERNAL F
        DIMENSION R(5,5)
        DATA ER,A,B,N/1.E-5,0.0,3.14159,5/
        CALL ROMBRG(F,A,B,N,R,ER)
        STOP
        END
```

```
0.3982251E-05
0.1570797E+01  0.2094395E+01
0.1896119E+01  0.2004560E+01  0.1998571E+01
0.1974232E+01  0.2000269E+01  0.1999983E+01  0.2000006E+01
0.1993570E+01  0.2000017E+01  0.2000000E+01  0.2000000E
  +01  0.2000000E+01
```

即

用龙贝格求积算法计算积分 $\int_0^{\pi} \sin x \mathrm{d}x$**,取** $N=5$ **时的结果列于附表 2.**

与积分的精确值 $\int_0^{\pi} \sin x \mathrm{d}x=2$ **相比较,可见龙贝格求积算法的加速效果是很明显的.**

附表 2 计算结果

0				
1.570797	2.094395			
1.896119	2.004560	1.998571		
1.974232	2.000269	1.999983	2.000006	
1.993570	2.000017	2.000000	2.000000	2.000000

实习题一

用龙贝格求积算法上机计算：

1. $\int_0^1 \frac{1}{1+x^2}\mathrm{d}x$；

2. $\int_0^{\pi} \mathrm{e}^x \cos x \mathrm{d}x$，误差限 $\varepsilon = 10^{-4}$；

3. $\int_1^3 3^x x^{1.4}(5x+7)\sin x^2 \mathrm{d}x$.

注：本题的数学模型背景：

20 世纪 70 年代以来，随着对柴油机、汽油机等缸内工作过程研究的进一步发展，国内外科技人员开始对其最主要、最复杂的燃烧过程运用数学方法来进行研究，譬如，建立单区、多区的数学模型，以达到最佳设计.

例如，较简单的单区模型累积放热量计算 Q，即抽象为本题的算例.

二、求解线性代数方程组的高斯-赛德尔迭代算法和超松驰迭代算法

1. 方法简介

给定线性代数方程组

$$\boldsymbol{AX} = \boldsymbol{b}$$

其中，$\boldsymbol{A} = (a_{ij})_{n\times n}$ 为系数矩阵，$\boldsymbol{b} = (b_1, b_2, \cdots, b_n)^{\mathrm{T}}$ 为右端向量，$\boldsymbol{X} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 为解向量，并设 $a_{ii} \neq 0$，$(i = 1,2,\cdots,n)$

用迭代法求解方程组 $AX=b$,先将它化为等价的方程组:

$$x_i=\frac{1}{a_{ii}}(b_i-\sum_{j\neq i}a_{ij}x_j),\quad (i=1,2,\cdots,n)$$

则 (1) 用高斯-赛德尔迭代法求解方程组的计算公式为

$$\begin{cases} x_i^{(k+1)}=x_i^{(k)}+\dfrac{1}{a_{ii}}(b_i-\sum\limits_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum\limits_{j=i}^{n}a_{ij}x_j^{(k)}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\begin{pmatrix} i=1,2,\cdots,n \\ k=0,1,2,\cdots \end{pmatrix} \\ \text{初值 } x_i^{(0)}(i=1,2,\cdots,n)\text{为已知}. \end{cases}$$

(2) 用超松驰迭代算法求解方程组的计算公式为

$$\begin{cases} x_i^{(k+1)}=x_i^{(k)}+\dfrac{\omega}{a_{ii}}(b_i-\sum\limits_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum\limits_{j=i}^{n}a_{ij}x_j^{(k)}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\begin{pmatrix} i=1,2,\cdots,n \\ k=0,1,2,\cdots \end{pmatrix} \\ \text{初值 } x_i^{(0)}(i=1,2,\cdots,n)\text{为已知}. \end{cases}$$

其中,ω 为松驰因子($0<\omega<2$).

显然,当 $\omega=1$ 时的 SOR 方法就是 G-S 迭代法,它是在 G-S 迭代法的基础上,引进一个加速参数 ω 而产生的一个加速迭代方法,对求解某些由偏微分方程数值解而产生的线性代数方程组特别有用.

但在使用上述迭代法前,一般应先判别该方程组是否满足收敛条件,例如有:

定理 若方程组 $AX=b$ 的系数矩阵 A 为对称正定阵,则当 $0<\omega<2$时,超松驰迭代法收敛.

2. 程序及例

由于 G-S 迭代法就是当 $\omega=1$ 时的 SOR 方法,下面只给出用 SOR 方法解方程组 $AX=B$ 的计算程序.

数组 X 为一组工作单元，开始存放初始向量，接着存放中间数据，最后存放结果，用

$$|p_0| = \max_{1\leqslant i\leqslant n} |\Delta x_i| = \max_{1\leqslant i\leqslant n} |x_i^{(k+1)} - x_i^{(k)}| < eps(\textbf{精度要求})$$

控制迭代终止，k **表示迭代次数**.

(1) **程序要点**

① $k \leftarrow 0$；

② $x_i \leftarrow 0 (i=1,2,\cdots,n)$；

③ $k \leftarrow k+1$；

④ $p_0 \leftarrow 0$；

⑤ **对于** $i=1,2,\cdots,n$

(a) $p \leftarrow \Delta x_i = \omega (b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i}^{n} a_{ij} x_j) / a_{ii}$；

(b) **若** $|p| > |p_0|$，**则** $p_0 \leftarrow p$；

(c) $x_i \leftarrow x_i + p$.

⑥ **若** $|p_0| > eps$，**则转**③；

⑦ **输出结果** x, k.

(2) **子程序名** SOR(A,B,X,N,OMGA,EPS,K)

(3) **计算实例**

求方程组 $\boldsymbol{AX}=\boldsymbol{B}$ **的解. 其中**

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 4 \\ -3 \end{bmatrix}$$

$\omega = 1.03$，**迭代次数** $k=6$

$\omega = 1$，**迭代次数** $k=7$

$\omega = 1.5$，**迭代次数** $k=21$

由此，选取合适的松弛因子，在同样精度下，可减少迭代次数.

(4) **程序**

```
SUBROUTINE SOR(A,B,X,N,OMGA,EPS,K)
DIMENSION A(N,N),B(N),X(N)
```

```
      K=0
      DO 1 I=1,N
      X(I)=0.0
1     CONTINUE
2     PO=0.0
      K=K+1
      DO 4 I=1,N
      S=0.0
      DO 3 J=1,N
      S=S+A(I,J)*X(J)
3     CONTINUE
      P=OMGA*(B(I)-S)/A(I,J)
      IF(ABS(P).GT.ABS(PO))THEN
      PO=P
      END IF
      X(I)=X(I)+P
4     CONTINUE
      IF(ABS(PO).GT.EPS)THEN
      GO TO 2
      END IF
      RETURN
      END

      DIMENSION A(3,3),B(3),X(3)
      DATA  A/4.0,-1.0,0.0,-1.0,4.0,-1.0,0.0,-1.0,4.0/,B/1.0,4.0,-3.0/
      DATA OMGA/1.03/,EPS/5.0E-6/
      CALL SOR(A,B,X,3,OMGA,EPS,K)
      OPEN(10,FILE='OU5.DAT')
      WRITE(10,*)'---------------A---------------'
      DO 2 I=1,3
2     WRITE(10M3)(A(I,J),J=1,3)
      WRITE(10,*)'---------------B---------------'
      WRITE(10,3)(B(J),J=1,3)
      WRITE(10,*)'---------------X---------------'
```

```
        WRITE(10,3)(X(J),J=1,3)
        WRITE(10,4)K
        WRITE(10,5)OMGA
3       FORMAT(5X,E14.8,5X,E14.8,5X,E14.8)
4       FORMAT(5X,'K=',5I)
5       FORMAT(5X,'OMGA=',E14.8)
        CLOSE(10)
        STOP
        END
```

```
---------------A---------------
         0.40000000E+01         -.10000000E+01          0.00000000E+00
        -.10000000E+01           0.40000000E+01         -.10000000E+01
         0.0000000E+00          -.10000000E+01           0.40000000E+01
---------------B---------------
         0.10000000E+01          0.40000000E+01         -.30000000E+01
---------------X---------------
         0.50000024E+00          0.10000001E+01         -.49999997E+00
         K=              6
        OMGA=0.10300000E+01


---------------A---------------
         0.40000000E+01         -.10000000E+01          0.00000000E+00
        -.10000000E+01           0.40000000E+01         -.10000000E+01
         0.00000000E+00         -.10000000E+01           0.40000000E+01
---------------B---------------
         0.10000000E+01          0.40000000E+01         -.30000000E+01
---------------X---------------
         0.50000048E+00          0.10000002E+01         -.49999994E+00
         K=              7
        OMGA=0.10000000E+01


---------------A---------------
         0.40000000E+01         -.10000000E+01          0.00000000E+00
        -.10000000E+01           0.40000000E+01         -.10000000E+01
```

0.00000000E+00 　　-.10000000E+01 　　0.40000000E+01

---------------B---------------

0.10000000E+01 　　0.40000000E+01 　　-.30000000E+01

---------------X---------------

0.49999937E+00 　　0.99999928E+00 　　-.50000024E+00

K= 　　21

OMGA=0.15000000E+01

实习题二

1. 分别用高斯-赛德尔迭代法与 $\omega=1.25$ 的超松驰迭代法解方程组

$$\begin{cases} 4x_1+3x_2 = 24 \\ 3x_1+4x_2-x_3=30 \\ -x_2+4x_3=-24. \end{cases}$$

2. 用高斯-赛德尔迭代法求解七元一次方程组

$$\begin{bmatrix} 2.418 & -1.061 & 2.669 & 4.361 & -0.119 & -1.209 & -0.500 \\ -1.501 & 19.832 & 0.694 & -4.816 & 2.274 & 2.001 & -1.909 \\ 2.308 & 1.728 & -15.165 & -2.023 & 1.104 & 2.107 & -1.000 \\ 3.359 & -0.913 & -6.441 & 27.864 & 3.737 & -4.375 & -2.375 \\ -1.562 & 1.168 & -2.004 & 1.818 & 9.490 & 0.401 & -1.073 \\ 1.174 & 7.318 & -2.278 & -0.143 & -9.835 & -31.670 & 4.114 \\ 0.109 & -1.313 & -0.900 & -1.972 & -3.514 & -1.107 & 12.094 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 8.262 \\ -33.818 \\ -52.673 \\ -97.284 \\ 20.351 \\ 149.9188 \\ 81.653 \end{bmatrix}.$$

注:本题的数学模型背景:

在用有限元方法求解平面的应力应变时,最后能得到一个线性代数方程组,$\boldsymbol{AX}=\boldsymbol{b}$,其中,系数矩阵 $\boldsymbol{A}$ 表示该实际问题经过约束处理后的总刚度矩阵,右端向量 $\boldsymbol{b}$ 是总荷载向量.如上述算例就是一个应用实例.

三、求解常微分方程初值问题的龙格-库塔方法

1. 方法简介

对于常微分方程初值问题

$$\begin{cases} \frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y) \\ y(x_0)=y_0 \end{cases}, \quad (a\leqslant x\leqslant b)$$

常用的是标准四阶龙格-库塔方法

$$\begin{cases} y_{n+1}=y_n+\frac{1}{6}(k_1+2k_2+2k_3+k_4) \\ k_1=hf(x_n,\ y_n) \\ k_2=hf\left(x_n+\frac{h}{2},\ y_n+\frac{1}{2}k_1\right) \\ k_3=hf\left(x_n+\frac{h}{2},\ y_n+\frac{1}{2}k_2\right) \\ k_4=hf\ (x_n+h,\ y_n+k_3), \quad (n=0,1,2,\cdots) \end{cases}$$

其中,h 为步长,使用者可根据题目的精度要求适当选择步长 h,由初值(x_0,y_0)出发,可得未知函数$y(x)$在区间$[a,b]$上相应各点处的数值解,其截断误差为$O(h^5)$.

2. 程序及例

(1) 程序要点

对初值问题

$$\begin{cases} x'=F(x,t) \\ x(0)=S \end{cases}$$

四阶龙格-库塔方法:

$$x(t+h)=x(t)+\frac{1}{6}(F_1+2F_2+2F_3+F_4)$$

其中

$$F_1=hF(t,x)$$

$$F_2=hF\left(t+\frac{h}{2},x+\frac{1}{2}F_1\right)$$

$$F_3 = hF\left(t+\frac{h}{2}, x+\frac{1}{2}F_2\right)$$

$$F_4 = hF(t+h, x+F_3)$$

（2）**子程序** RK4(F,T,X,H,NSTEP)**中变量说明**

F——**初值问题的导函数,以函数子程序给出**；

T——**自变量的值**；

X——T **处的函数值**；

H——**步长**；

NSTEP——**计算步数**.

（3）**计算实例**

$$\begin{cases} x' = 2+(x-t-1)^2 \\ x(1) = 2 \end{cases}$$

在[1,1.5625]**上的值,步长为** H=1/128,**步数为** NSTEP=72.

（4）**程序(例子的结果是每两步打印一个值)**

```
SUBROUTINE RK4(F,T,X,H,NSTEP)
OPEN(1,FILE='OUT.DAT')
WRITE(1,3)T,X
H2=0.5*H
START=T
DO 2 K=1,NSTEP
F1=H*F(T,X)
F2=H*F(T+H2,X+0.5*F1)
F3=H*F(T+H2,X+0.5*F2)
F4=H*F(T+H,X+F3)
X=X+(F1+F2+F2+F3+F3+F4)/6.0
T=START+H*REAL(K)
IF(MOD(K,2).EQ.0)THEN
WRITE(1,3)T,X
END IF
```

```
2       CONTINUE
        CLOSE(1)
3       FORMAT(5X,2E22.14)
        RETURN
        END

        EXTERNAL,F
        DATA T/1.0/,X/2.0/,H/7.8125E-3/,NSTEP/72/
        CALL RK4(F,T,X,H,NSTEP)
        STOP
        END

        FUNCTION F(T,X)
        F=2.0+(X-T-1.0)**2
        RETURN
        END
```

```
0.10000000000000E+01    0.20000000000000E+01
0.10156250000000E+01    0.20312514305115E+01
0.10312500000000E+01    0.20625104904175E+01
0.10468750000000E+01    0.20937848091125E+01
0.10625000000000E+01    0.21250820159912E+01
0.10781250000000E+01    0.21564097404480E+01
0.10937500000000E+01    0.21877760887146E+01
0.11093750000000E+01    0.22191886901855E+01
0.11250000000000E+01    0.22506554126740E+01
0.11406250000000E+01    0.22821846008301E+01
0.11562500000000E+01    0.23137843608856E+01
0.11718750000000E+01    0.23454627990723E+01
0.11875000000000E+01    0.23772287368774E+01
0.12031250000000E+01    0.24090907573700E+01
0.12187500000000E+01    0.24410574436188E+01
0.12343750000000E+01    0.24731380939484E+01
0.12500000000000E+01    0.25053420066833E+01
```

0.12656250000000E+01　　0.25376787185669E+01
0.12812500000000E+01　　0.25701582431793E+01
0.12968750000000E+01　　0.26027905941010E+01
0.13125000000000E+01　　0.26355862617493E+01
0.13281250000000E+01　　0.26685562133789E+01
0.13437500000000E+01　　0.27017116546631E+01
0.13593750000000E+01　　0.27350642681122E+01
0.13750000000000E+01　　0.27686264514923E+01
0.13906250000000E+01　　0.28024106025696E+01
0.14062500000000E+01　　0.28364300727844E+01
0.14218750000000E+01　　0.28706984519958E+01
0.14375000000000E+01　　0.29052300453186E+01
0.14531250000000E+01　　0.29400401115417E+01
0.14687500000000E+01　　0.29751441478729E+01
0.14843750000000E+01　　0.30105590820313E+01
0.15000000000000E+01　　0.30463023185730E+01
0.15156250000000E+01　　0.30823919773102E+01
0.15312500000000E+01　　0.31188471317291E+01
0.15468750000000E+01　　0.31556885242462E+01
0.15625000000000E+01　　0.31929373741150E+01

实习题三

用标准四阶龙格-库塔方法求解下列常微分方程初值问题：

1. $\begin{cases} y'=y-\dfrac{2x}{y}, & \text{取 } h=0.2 \\ y(0)=1, & (0\leqslant x\leqslant 1) \end{cases}$；

2. $\begin{cases} y'=-y+x+1, & \text{取 } h=0.1 \\ y(0)=1, & (0\leqslant x\leqslant 1) \end{cases}$.

附录Ⅱ　数学软件在“数值计算基础”课程中的应用

随着数学计算机硬件技术的飞速发展,一大批功能强大的数学软件系统应运而生．Mathematica 与 MATLAB 是其中两个具有代表性而且被广泛应用的数学软件．其中,Mathematica 具有浓郁的数学风味,它拥有强大的数学公式编辑、符号计算和图形化的功能．尤其是自 3.0 的版本开始,Mathematica 提供了称为笔记本(Notebook)的运算界面前端,用户可以方便地在其中书写数学公式、编写程序、显示图形,而 Mathematica 的内核(Kernel)部分则负责实行数学运算和代码的执行．Mathematica 虽然有众多复杂的内部函数调用,要结合相当的数学背景知识才能了解,但是在 Mathematica 中,不论是简单的四则运算,还是复杂的微分方程求解,其输入参数、输出结果都统一在一个结构模式下,那就是“表达式”．因此,初学者只要掌握了表达式的概念,就比较容易掌握 Mathematica 的用法．

MATLAB 则更具有工程应用的风格．它是从一个数学教学软件发展而来的．由于它处理向量和矩阵运算便捷,最早被应用于求解控制工程领域的数学问题．此后,MATLAB 不断以工具箱(Toolbox)的形式添加应用于某一特定工程领域的函数包．由于它在可扩充性方面的强大优势,不断有新的工具箱被开发并且添加到 MATLAB 系统中．因此,它总是紧跟科研和工程界的新领域和新技术．这也使得它被广泛地使用．

由此可以看到,Mathematica 和 MATLAB 是两个庞大的软件系统．希望读者从这个附录中对这两个软件系统有一个总体的了解．并且附录将重点地介绍这些数学软件在“数值计算基础”课

程中在解答习题方面的应用及有关的具体实例．可以说，“数值计算基础”的课程为我们提供了这样一个边学边用的机会．希望读者在课程进行中，在学习数学知识的同时，学会使用这些数学软件，而使用这些数学软件，又可以帮助读者深入了解所学的数学知识，起到相辅相成的作用．

一、Mathematica 使用初步

1. Mathematica 简介

Mathematica（以下简称麦卡）是一个能够进行数学运算的计算机软件．它结合了符号演算和数值计算的功能于一身，在科学与工程的各个领域有着广泛的应用．从符号演算方面的功能来看，麦卡的主要设计者 Stephen Wolfram 最初就是希望麦卡的出现能帮助他解决粒子物理研究中的繁复的公式演算工作．因此，麦卡具有与身俱来的符号演算方面的强大功能，并不弱于同时代的计算机符号演算系统 Maple. 另一方面，麦卡在数值计算方面的表现也可以和工程计算中流行的 MATLAB 系统相匹敌．从计算机编程的方面来看，麦卡提供了一种全新的编程思维方式．一般，将数值计算的理论应用于解决实际问题时，往往要在计算机上编程和运算，再得出有意义的结果．传统的方法是通过 FORTRAN，C 等计算机语言编写程序，然后上机编译、运行和调试，得出计算结果，最后结合实际问题分析结果的合理性．而麦卡是一个集成的解释运行系统，用户只要在一个工作区内（往往是一个类似于文本编辑器的输入窗口）输入命令或代码，系统可以立即执行，得出结果，而不必经过编译的过程．这样的解释执行系统是硬件速度的提高和软件系统向高层次发展的必然结果．它对于编程思维方式的改进是巨大的．下面先举一个简单的例子：

例 1 计算阶乘 n！

（1）类似于 C 语言中的递归函数，麦卡也可以定义这样一个

递归函数来求阶乘．

输入 factorial [n _]:=factorial[n-1]n;
factorial[1]=1;

可以看到，在麦卡中，函数的参数是写在一个方括号[ ]内的，还有，乘法运算也不必用 * 来表示，这一点和我们平时手写数学表达式的习惯很相似．以上两句语句就定义了一个求阶乘的函数 factorial．例如，我们可以试试求 6 的阶乘：

输入：factorial[6]
输出：720

如果你是第一次在 Windows 系统中使用麦卡 3.0 或 4.0 的系统，请记住要执行一个命令或代码时，在行末同时按 Shift 和 Enter 两个键．

（2）除了以上这种自己编程的方式，还可以直接调用麦卡的内部函数．

输入：Factorial[6]
输出：720

所得的结果是相同的．可以发现第一次求阶乘调用的是 factorial，函数开头的第一个字母 f 是小写的，这样就调用了我们自己编写的递推函数．而第二次调用的 Factorial，开头字母 F 是大写的，这样就调用了麦卡的内部函数．可见麦卡对大小写是敏感的．实际上，麦卡就是这样规定的，内部函数名的第一个字母全是大写的．为了不使函数名冲突，我们则尽可能用小写字母来命名自定义函数．

（3）当然，要计算 6 的阶乘，还有更简便的方法，只要输入 6！加上 Shift 和 Enter 键，结果就出来了．

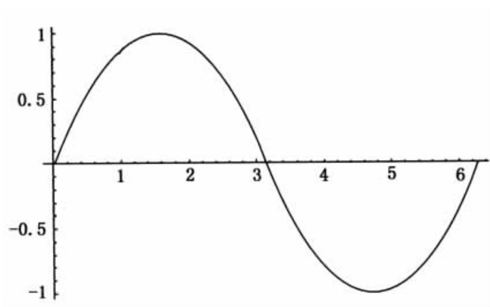
输入：6！
输出：720

从以上三种计算阶乘的方式中可以看到，麦卡是一个高效的数学软件系统．尤其是它的编程和使用风格，与我们的数学习惯很接近，因此，在研究和教学领域，麦卡具有独特的优势．

但要全面地了解和熟练地运用这一庞大的软件系统，则不是一蹴而就的事，好在一旦了解了该系统的基本思想和使用方法，再依照自己的兴趣和要解决的实际问题深入了解其中的某个部分就很容易了．值得指出的是，从麦卡 3.0 开始，麦卡的在线帮助中包括了 Stephen Wolfram 写的《Mathematica Book》，这是学习麦卡的一本很好的参考书．国内不少作者出版的介绍麦卡的书籍大多是部分翻译了 Stephen 的那本著作．读者可参考那些"中文版"的《Mathematica Book》．同时，在计算机上使用麦卡时，不时地查阅 Stephen 的英文原著，可学会不少东西和解决实际中遇到的问题．此外，麦卡的网站（www. wolfram. com）和用户新闻组提供了许多有价值的资料，为深入学习和使用该系统提供了很多支持．

（1）数值计算和符号运算

麦卡能计算在科学计算器上遇到的所有运算，还包括函数作图（附图 1）．



附图 1

例如

输入：2＋5^20＋8！

输出: 95367431680947

输入: $\mathbf{Sin}\left[\frac{\pi}{4}\right]$

输出: $\frac{1}{\sqrt{2}}$

输入: $\mathbf{Plot[Sin[x],\{x,0,2\pi\}]}$

输出:

输入: $\frac{\mathbf{1}}{\mathbf{2}}+\frac{\mathbf{1}}{\mathbf{3}}$

输出: $\frac{5}{6}$

注意到了上式中两个分数的和,结果还是一个分数.这就是麦卡的符号运算的魅力,它通过表达式的转换和无限精度运算的算法,从而得到化简了的符号解.尤其是表达式转换的功能使得麦卡能够非常接近真正的数学思维.当然,也可以要求得出表达式的数值解,那只要

输入: $\mathbf{N}\left[\frac{\mathbf{1}}{\mathbf{2}}+\frac{\mathbf{1}}{\mathbf{3}}\right]$

输出: 0.833333

其中, N **是一个麦卡的内部函数,表示求出表达式的数值解.**
和传统的计算机语言一样,麦卡中也可以定义变量,并给它们赋值.例如:

c=5

5

b+cx

b+5x

由此可见,一旦 c **被赋值之后,其后出现的表达式中的** c **的值就会被用** 5 **这个数字代替,而符号** b **由于没有被赋值,仍然保留在表达**

式中．当然，如果希望 c 还是以符号的形式回到表达式中，可用

Clear[c]

命令，将 c 的赋值去掉，这样再运行

b+cx

就会得到

b+cx

再来看一个解方程的例子：

Solve[x^2－5x+6==0,x]

{{x→2},{x→3}}

这里出现的大括号{}将在后面表达式一节介绍．实际上，对于一般的二次方程，麦卡也能推出其求根公式：

Slove[ax^2+bx+c==0,x]

$$\left\{\left\{x\to\frac{-b-\sqrt{b^2-4ac}}{2a}\right\},\ \left\{x\to\frac{-b+\sqrt{b^2-4ac}}{2a}\right\}\right\}$$

对于一个方程，也可以用一个变量来命名，例如：

eqn1=x^2－3y==3；eqn2=x+y==2；

上面的表达式中，后面是连续输入两个等号==，表示组成一个方程，而前面的一个等号就是变量的赋值了．

Solve 是麦卡中一个求解方程的函数，当然，麦卡中有上百个各种各样的函数，如何知道其调用规则呢？你可以在系统的在线帮助中查询一个函数的用法，另一个方法是在你的命令窗口内直接输入：

?Solve

Solve [eqns，vars] attempts to solve an equation or set of equa-

tions for the variables vars.

Solve [eqns, vars, elims] attempts to solve the equations for vars, eliminating the variables elims.

系统会立即给出该函数的用法．读者可试着查询一下刚才用过的函数 Plot **的具体用法．**

现在，用 Slove **求解刚才输入的两个方程：**

Solve[{**eqn1**，**eqn2**}，{**x**，**y**}]

$$\left\{\left\{y\to\frac{7}{2}-\frac{3\sqrt{5}}{2},\ x\to\frac{3}{2}(-1+\sqrt{5})\right\},\right.$$

$$\left.\left\{y\to\frac{1}{2}(7+3\sqrt{5}),\ x\to\frac{3}{2}(-1-\sqrt{5})\right\}\right\}$$

值得指出的是，符号解不是万能的．例如，对于以下的方程，麦卡就不能求得符号解：

Solve[$\mathbf{ax^5+bx^4+cx^3+dx^2+ex+d==0,x}$]

实际上，通过伽罗华理论已经证明高于四次的一般代数方程没有代数解，麦卡当然也就构造不出这样的解来．此外，还有不少问题，用无限精度的符号解法会使运算的复杂程度和所花费的运算时间达到无法忍受的程度，对这样的问题，我们往往选择其数值解．因此，麦卡还有一个数值求解方程的内部函数 FindRoot**，在求解非线性方程的根时，这个函数也很有用．在学习了数值计算的基础课程后，你也可以试着编写自己的求根函数．**

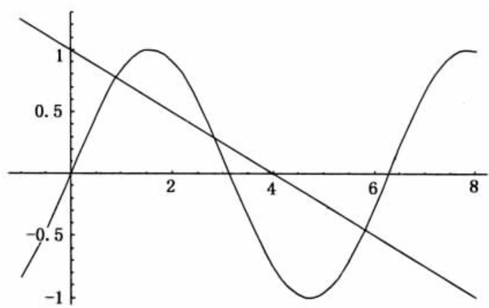
例如要求解

$$\mathbf{eqn=Sin[x]==1-\frac{x}{4};}$$

这样，一个方程的根，先用 Plot **函数将** Sin(x)**和** 1－x/4 **在直角坐标系内作图．**

$$\text{Plot}\left[\left\{\sin[x], 1-\frac{x}{4}\right\}, \{x, -1, 8\}\right]$$

如果你已经查询过了麦卡关于 Plot 函数的在线帮助,不难理解上面这个语句的调用.



附图 2

从附图 2 中可见,方程有 3 个根. 我们先来求一个离 1 最近的根:

```
FindRoot[eqn,{x,1}]
{x→0.890487}
```

变量 eqn 已在前面定义为所要求解的方程. 作为练习,读者能求得方程的另外两个根吗(先用?FindRoot 查一下该函数的用法)?

从以上这些简单例子可以看到,麦卡可以进行数值运算,也可以进行符号运算. 它的变量不需要事先说明类型,就可以直接赋值和使用. 各种内部函数的调用也很方便. 而贯穿其中的一个表达式的概念相当重要. 下一节我们就将介绍表达式的概念.

(2) 表达式运算

在麦卡中到处可见表达式,它的概念非常重要. 表达式的基本构造形式是 $f[a_1, \cdots, a_k]$,其中,f 是表达式的头,方括号[]内 a_1 到 a_k 是它的元素. 表达式还可以嵌套,例如:f[1,g[2]]是一个表

达式，这时，g[2]是 f 为头的表达式的元素，而 g[2]本身也是一个只有一个元素 2 的表达式．

麦卡中的表达式和我们日常说的数学表达式有什么关系呢？例如：

```
2+3
```

显然是一个数学表达式，在麦卡中，它实际上等价于

```
Plus[2,3]
```

这就是麦卡的表达式，它是由一个表达式头，加上后面的方括号内的参数构成的．在上面的例子中，Plus 就是一个表达式头，而方括号内的 2，3 就是其参数．这样，麦卡中的表达式实际上和内部函数的调用是一个概念．一个复杂的表达式实际上和一组相互嵌套的函数调用是等价的．例如：

```
FullForm[(a+b)c]
Times[Plus[a,b],c]
```

表达式的嵌套在上面出现了．其中，函数 FullForm 就是求取一个表达式在麦卡内的形式．上式中出现了圆括号()，它和我们在数学中遇到的圆括号的概念相同，表示运算次序的先后．之所以麦卡中函数调用使用方括号，就是为了不和数学的方式相冲突．

```
d(2+3)
5d
FullForm[%]
Times[5,d]
d[2+3]
d[5]
FullForm[%]
d[5]
```

从上面的两段代码可以看出方括号和圆括号在麦卡中的区别．另外，FullForm 函数调用的参数％表示取上次计算的结果作为函数的输入参数．

了解了方括号、圆括号的用法，再说明一下大括号｛　｝的用法：

例如：

```
{1,2,3,4}
```

也是麦卡中的一个表达式．调用 FullForm，即

```
FullForm[{1,2,3,4}]
List [1,2,3,4]
```

也即，大括号可以括起一个列表．如果列表中都是数字，就组成了一个数组，这里的数组的概念和传统的计算机语言中所说的数组是一样的．只不过在麦卡中，它的实现更方便，功能更强大．例如，用内部函数 Table 可以构造一个平方数表：

```
Table[i^2,{i,1,10}]
{1,4,9,16,25,36,49,64,81,100}
```

而数组的数组，即二维数组就可以组成矩阵：

```
Table[Random[],{3},{2}]
{{0.768709,0.247905},{0.681973,0.768977},
{0.735583,0.169711}}
MatrixForm[%]
```

$$\begin{bmatrix} 0.768709 & 0.247905 \\ 0.681973 & 0.768977 \\ 0.735583 & 0.169711 \end{bmatrix}$$

这里，Table 函数产生了一个由随机数组成的 3×2 的矩阵．MatrixForm 只是将所产生的二维数组表示成矩阵形式．当然，除

了 Table 函数可以建立列表之外，还有许多内部函数可以完成类似的工作，如 Array，Range，DiagonalMatrix，IdentityMatrix 等，读者可以分别查询它们的在线帮助．

麦卡定义了这些数组、矩阵的表示，当然也有对它们进行运算、操作的函数．例如求一个矩阵的特征根：

$$\textbf{Eigenvalues}\left[\begin{pmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ 0 & 1 & 5 \end{pmatrix}\right]$$

$\{5, -\sqrt{7}, \sqrt{7}\}$

又例如取数组的某个元素：

data＝{10,20,30,40,50}；

data[[3]]

data＝…

30

从上面的例子可以看到，麦卡的数组是从 1 开始索引的，而不像 C 语言那样从 0 开始索引数组．因此，data[[3]]取到的是数组中的第三个元素．选取列表中的元素还有如 Take，Drop，Select，Cases 等内部函数．读者需要时可详细查询其用法．

与 MATLAB 中的情况一样，麦卡中的不少初等函数可以对列表中的各个元素求值，结果还是一个列表，例如：

a＝Range[10]

{1,2,3,4,5,6,7,8,9,10}

Sin[a]

{Sin[1]，Sin[2]，Sin[3]，Sin[4]，Sin[5]，
Sin[6]，Sin[7]，Sin[8]，Sin[9]，Sin[10]}

现在从列表回到更广义的表达式的概念来看一下我们刚定义的变量 a，

FullForm[a]

List[1,2,3,4,5,6,7,8,9,10]

对于这个表达式,内部函数 Apply **可以将原表达式头** List **替换成** Plus,**这样,调用**

Apply[Plus,a]

55

就相当于调用:

Plus[1,2,3,4,5,6,7,8,9,10]

就得出了 55 **这个结果. 当然,要在麦卡中求这十个数字的和还有许多方法,例如:**

Sum[i,{i,10}]

55

对表达式运算的函数还有很多,我们将在后面用到时再做相应的解释.

(3) **函数定义**

从前面的介绍中我们已经看到了许多麦卡中函数调用的例子. 除了调用麦卡的内部函数,还可以定义自己的函数,这就是麦卡中的编程了. 在介绍编程之前,先要了解麦卡的"模式"(pattern)**的概念.**

模式是麦卡中广泛应用的一个很有用的概念. 例如:

f[x _]

就表示一个以 f **为表达式头的,含有一个元素的表达式,而**

f[1+π]

就属于这样的表达式,我们就称这个表达式是匹配的. 表达式匹配的目的就是可以将表达式的各个部分用变量来命名. 上面的例

子中，x 就代表了 1+π. 表达式匹配的直接作用就是可以进行表达式的替换，例如：

```
f[1+π]/. f[x_]→g[0,x]
g[0,1+π]
```

以上"/."表示替换，和内部函数 Replace 等同，其后就是替换的规则，箭头是通过输入"-"和">"两个符号组成的．也就是将表达式头为 f、含有一个元素的表达式替换为表达式头为 g 含有两个元素的表达式，其中第一个元素为 0，第二个元素是原来 f 表达式中的元素．

明白了模式和替换规则，就可以来定义函数了．例如，先来定义一个求平方的函数

```
f[x_]=x^2;
f[6]
36
```

对于 FullForm，可查看函数 f 的结构．例如：

```
Clear[f]
FullForm[x[x_]]
f[Pattern[x,Blank[]]]
```

可见，f 为头的函数，其参数是一个可以被匹配的模式．当调用这个函数时，f[6]中的 6 就和函数定义中的 x 匹配，从而计算了 6 的平方的值．

但如果只是定义

```
f[x]=x^2
```

x^2

f[6]就不会被计算出来，这样，也就不能称为函数定义了．

对于复杂的函数定义，往往要用到许多局部变量或内部变量，

这时 Module 和 Block 两个函数就很重要了．例如，定义一个函数

```
f[p_]:=(x=Cos[p];x/(1+x))
f[1]
```

$$\frac{\text{Cos}[1]}{1+\text{Cos}[1]}$$

```
?x
Global x
x=Cos[1]
```

这时，变量 x 是全局变量，并且已被赋值为 Cos(1)．如果希望 x 只是出现在函数 f 中，不会影响全局的变量，也不会被外界的操作所影响，就要使用 Module 了．

```
Clear[x]
f[p]:=Module[{x},x=cos[p];x/(1+x)]
f[1]
```

$$\frac{\text{Cos}[1]}{1+\text{Cos}[1]}$$

```
?x
Global x
```

这时，再运行 f[1]，虽然函数模块中的 x 被赋值，函数外的全局变量 x 还是不受影响．如果想要不光把函数中的变量命名空间保护起来，而且还要把函数名也保护起来，那就要考虑使用包(package)的方法，这里不作介绍．

2．用 Mathematica 计算《数值计算基础》的部分习题解答

这里的“数值计算基础”课程的习题选自本书作者沈剑华编写的《数值计算基础》(同济大学出版社)一书的课后部分习题．用麦卡完成课后习题有以下几个优点：

(1) 提高效率

由于习题中的计算量往往很大，用便携式的科学计算器来进行运算会耗费不少时间．尤其是当因繁复的计算出现错误时，不可避免地要进行一遍又一遍的重复计算．而麦卡具有科学计算器的全部功能，它明晰的数学表达式的输入和编辑方式，立即得出计算的结果，可以将运算的全过程表示和记录下来，即使出错了，也很容易修改错误的地方，不必全部从头重来．

（2）编程方便，代码的可读性强

从前面对于麦卡的简介可以看到，麦卡的编程方式独具特色．对于数值计算中的公式和算法几乎可以直接对应于麦卡程序的每一条语句．因此，用麦卡编写的程序可读性很好，修改和检错十分方便．尤其是麦卡是一个解释执行的系统，编写的代码不必经过编译就可运行，十分便捷．

（3）可以进行算法的分析和实验

虽然数值计算中的许多算法在麦卡中都有现成的内部函数可直接调用．但我们的习题解答不光是给出这些麦卡的函数，让读者调用一下，求出答案了事．我们试图根据数值计算基础课程中的基本理论和方法，搞清求解的来龙去脉，并尽量编写出实现同样功能的自定义函数．也许我们的自定义函数计算效率不及麦卡的内部函数，或者功能没有那么强大，然而对于某个算法，只有通过自己编写程序，进行算法实验，才能有较深刻的了解，对于使用这一算法时可能出现的问题才有可能解决．

在“数值计算基础”课程中出现的许多算法，虽已经有现成的C，Fortran语言的库函数可供调用，实际的工作中也不一定要自己编程去实现已有的函数．但要将数值算法有效正确地应用到实际问题中，并最终得到有用的计算结果，不是只靠知道几个函数调用规则，或是了争几个麦卡的内部函数所能够解决的．所以，在学习“数值计算基础”课程期间，麦卡不是用来“偷懒”的工具，而应该成为一个学习的辅助工具．

注：在习题解答中出现的麦卡系统的一些规则和用法，会在相应习题后面的“注”

中说明，这样，不影响解题思路的连续性，而对于不熟悉麦卡系统的读者，则可参照注解中的说明同时学习麦卡的使用方法．读者应试着从一些简单的课后练习做起，最终学会运用麦卡求解数值计算中的问题．此外，在习题解答中出现了许多麦卡内部函数的调用，希望读者能够及时查询麦卡的在线帮助，了解这些函数的用法．为了方便读者，我们在后面的"麦卡函数调用说明"一节中列出了其中的一些函数，并做了中文的说明，对于暂时没有麦卡系统的读者，可供参考．

（Ⅰ）习题二

1．已知 $f(0)=1, f(1)=2, f(2)=4$，求 $f(x)$ 的二次插值多项式．

解　调用麦卡的内部函数，可得

f = InterpolatingPolynomial[{{0,1},{1,2},{2,4}},x]

$1+\left(1+\frac{1}{2}(-1+x)\right)x$

验证一下这个多项式是否满足插值条件：

令 x=1 代入多项式

f/. x→1

2

与已知条件相符．

注：/. 表示一个替换规则，x→1 表示将多项式中的 x 用整数 1 代替．读者可试着将 x 用 0 和 2 代入，看是否符合已知条件．

2．给定函数 $y=\sin(x)$ 的函数表如下，试分别用线性插值与二次插值求 sin0.57891 的近似值，并估计截断误差．

TableForm[a={{0.4,0.5,0.6,0.7},
{0.38942,0.47943,0.56464,0.64422}}]

0.4	0.5	0.6	0.7
0.38942	0.47943	0.56464	0.64422

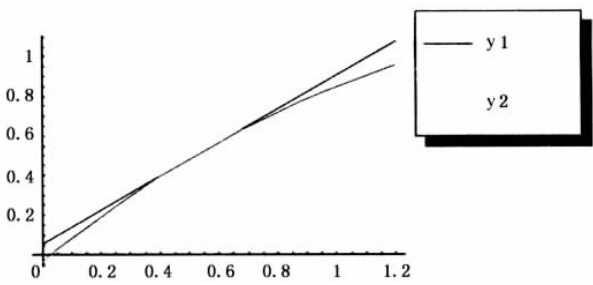
解　先求线性插值，选取与 0.57891 最接近的两个已知数据点

```
Transpose[a][[{2,3}]]
{{0.5,0.47943},{0.6,0.56464}}
y1 = InterpolatingPolynomial[%,x]
0.47943+0.8521(-0.5+x)
```

再求二次插值，同样的，选取接近 0.57891 的三个数据点

```
Transpose[a][[{2,3,4}]]
{{0.5,0.47943},{0.6,0.56464},{0.7,0.64422}}
y2 = InterpolatingPolynomial[%,x]
0.47943+(0.8521-0.2815(-0.6+x)(-0.5+x))
```

我们还可以画出两个插值多项式的图形（题图 1）



题图 1

```
《Graphics\Legend\
Plot[(y1,y2},{x,0,1.2},]
PlotStyle→{{RGBColor[1,0,0]},{RGBColor[0,1,0]}},
PlotLegend→{"y1","y1"},LegendPosition→{1,0.1}]
```

sin0.57891 的值和插值函数 y1，y2 求得的值的比较（取十位有效数字）：

Map[NumberForm[#, 10]&, {sin[0.578 91], y1, y2}/. x→0.57891]

{0.5471118671,0.546669211,0.5471376866}

现在根据公式自定义函数来估计截断误差：

resError[x_,n_],xp_List,s′_]：=

$$\frac{D[Sin[t],\{t,n+1\}]}{(n+1)!}\prod_{j=0}^{n}(x-xp)[[j+1]])/.\ t\to s'$$

resError[0.57891,1,{0.5,0.6},0.6]

0.000469842

resError[0.57891,2,{0.5,0.6,0.7},0.5]

−0.000029475

从截断误差可以看出，线性插值只能达到3位有效位数的精度，而二次插值就有4位有效位数的精度.

注：Transpose**是麦卡的内部函数，它只是将数据矩阵**a**转置一下.**A[[{2,3,4}]]**表示取序列**A**中的第二，三，四个元素. 符号**%**则表示上一步的计算结果，可用来代入下一步的运算中，**Plot**函数调用时多了一些选项，主要是要将一次和二次插值多项式的函数图形用两种不同的颜色显示出来.**NumberForm**是按精度要求显示出计算结果的有效位数.**NumberForm[Sin[0.57891],10]**就是将结果显示**10**位有效数字.**Map**函数见"三".**

4. **假设对函数** $f(x)$ **在步长为** h **的等距点上造表，且** $|f''(x)|\leqslant M$**，证明：在表中任意相邻两点间做线性插值，误差不超过** $\frac{1}{8}Mh^2$.

设 $f(x)=\sin x$**，问** h **应取多大才能保证线性插值的误差不大于** $\frac{1}{2}\times10^{-6}$.

解　先定义线性插值的截断误差函数

resError[x_]= $\frac{1}{2!}$**x(x−h)**

求以上函数取最大值的 x

```
Solve[D[resError[x],x]==0,x]
```

$\left\{\left\{x\to\frac{h}{2}\right\}\right\}$

代回函数,可得最大值

$$\mathbf{resError[x]/.\ x\to\frac{h}{2}}$$

$-\frac{h^2}{8}$

再对函数 f(x)=sin(x),**在误差范围内,求满足要求的步长** h

```
<<Algebra\InequalitySolve\
```

$$\mathbf{InequalitySolve\left[\frac{h^2}{8}\leqslant\frac{1}{2}*10^{-6},h\right]//N}$$

$-0.002\leqslant h\leqslant 0.002$

取正的部分,所以步长 h **不应大于** 0.002,**以保证误差在题设的范围内.**

注:Solve **函数已经在麦卡简介中介绍过了.** D **是麦卡中求函数微分符号解的内部函数. 例如**,D[Cos[x],x]**得出**-Sin[x]. **本题中,令误差函数的导数为零,求得其极值点.** <<Algebra\InequlitySolve\ **表示载入一个求解不等式的函数包,这类似于在** C **语言中用** include **语句包含函数定义的头文件. 只有载入了这个函数包**(package),**其中的函数** IneqalitySolve **才能被调用来求解不等式.**

6. **根据下列函数表,作** 4 **次牛顿插值多项式,并求** $x=1.05$ **时** $\ln x$ **的近似值.**

x	1.00	1.02	1.06	1.08	1.09
$y=\ln x$	0.00000	0.01980	0.05827	0.07696	0.08618

解　根据差商的定义,可自定义求差商表的函数

```
diffNewton[inputList _]: =Module[{myDiffernce},
myDifference[{a1 _,a2 _}]: = (a1[[2]]-a2[[2]])/(a1[[1]]-a2[[1]]);
myDifference[{a1 _,r _ _,a3 _}]: =
(myDifferece[{a1,r}]-myDifference[{r,a3}])/(a1[[1]]-a3[[1]]);
myDifference[inputList]]
```

其中,myDifference 对应于差商的递推公式. 由已知,定义数据表 data:

```
data={{1,0},{1.02,0.0198},{1.06,0.05827},
{1.08,0.07696},{1.09,0.08618}};
```

现在就可以求各阶差商了

```
Table[Map[diffNewton,Partition[data,i,1]],{i,2,5}]//
TableForm
```

0.99	0.96175	0.9345	0.922
-0.470833	-0.454167	-0.416667	
0.208333	0.535714		
3.63757			

取差商表的第一列,就可构成牛顿插值多项式:

```
newtonexpression=
0+0.99(x-1)-0.470833(x-1)×(x-1.02)+
0.208333(x-1)(x-1.02)(x-1.06)
+3.63757(x-1)(x-1.02)(x-1.06)(x-1.08)//Expand
```

$2.55208-13.7523x+22.4866x^2-14.924x^3+3.63757x^4$

上述数值和麦卡的内部函数 InterpolatingPolynomial 产生的牛顿插值多项式比较,其结果就该是相同的.

```
InterpolatingPolynomial[{{1,0},{1.02,0.0198},{1.06,0.05827},
```

```
{1.08,0.07696},{1.09,0.08618}},x]//Expand
```

$2.55207-13.7523x+22.4866x^2-14.9239x^3+3.63757x^4$

由于该多项式是对自然对数函数的逼近,取 x=1.05 时的值

```
newtonexpression/.x→1.05
0.0487923
```

可以和 ln(1.05)比较

```
Log[1.05]
0.0487902
```

可见近似值的有效位数有 5 位.

注:此题中 diffNewton 是一个自定义函数.在麦卡简介中,我们介绍过在麦卡中定义函数的基本方法.从这个例子中,我们可以看到,Module 模块把 myDifference 隐藏在其内部,相当于面向对象编程中类的私有成员函数.和面向对象技术中函数的多态性概念一样,myDifference 可以根据不同的输入参数定义为多个函数.这样定义递归函数就很方便,也符合数学定义的方式.不必像 C 语言那样只能定义一个递归函数,而要在递归函数内加分支处理的判断语句.Partition 这个函数很关键,见附录.TableForm 只是将计算结果表示成一个表格的形式.

7. 已知函数 $y=f(x)$的函数表如下:

x	0.0	0.1	0.2	0.3	0.4	0.5
$f(x)$	1.00	1.32	1.68	2.08	2.52	3.00

根据函数表,求向前差分表,并写出牛顿向前插值公式.

解 先定义函数表

```
data={{0.,1.},{0.1,1.32},{0.2,1.68},{0.3,2.08},
  {0.4,2.52},{0.5,3.}};
```

只要修改一下上一题定义的差商函数 diffNewton,就可以求差分表

```
deltaF[inputList_]:=Module[{myDifference},
  myDifference[{a1_,a2_}]:=a2[[2]]-a1[[2]];
```

```
myDifference[{a1 _,r _ _,a3 _}]: =
myDifference[{r,a3}]-myDifference[{a1,r}];
myDifference[inputList]]
Table[Map[deltaF,Partition[data,i,1]],{i,2,6}]//
TableForm//Chop
```

0.32	0.36	0.4	0.44	0.48
0.04	0.04	0.04	0.04	
0	0	0		
0	0			
0				

由各阶向前差分表，可得出牛顿向前插值多项式：

$$\text{expr}=1+0.32\text{t}+0.04\frac{\text{t}(\text{t}-1)}{2}//\text{Expand}$$

$1+0.3\text{t}+0.02\text{t}^2$

若上式中有 x=0.0+0.1t，将 t 用 x 代回，就可和麦卡内部函数得出的结果比较一下

```
expr/.t→10x
```

$1+3.\text{x}+2.\text{x}^2$

```
InterpolatoingPolynomial[data,x]//Expand//Chop
```

$1.+3.\text{x}+2.\text{x}^2$

和麦卡的结果完全一样.

注：Chop 是将计算结果中接近 0 的表达式，用 0 代替. 读者可试着去掉上面一句命令中的 Chop，看会有什么结果. 另外，// 符号表示麦卡中的后缀表达式，也就是说，// 前面的表达式是 // 后面的函数调用的参数. 例如，Sin[1]//N 等价于 N[Sin[1]].

8. 试用两种不同的插值方法求一个三次多项式 $P(x)$，使其在节点 $x_0=0$，$x_1=1$ 上满足条件：

$$P(0)=f(0)=0,\qquad P(1)=f(1)=1,$$
$$P'(0)=f'(0)=-3,\quad P'(1)=f'(1)=9$$

其中 $P'(x)$，$f'(x)$是 $P(x)$，$f(x)$的一阶导数.

解法 1

```
InterpolatingPolynomial[{{0,{0,-3}},{1,{1,9}}},
x]//Expand
-3x+4x^3
```

解法 2

```
suppose=ax^3+bx^2+cx+d
d+cx+bx^2+ax^3
Solve[{(suppose/. x→0)==0,(suppose/. x→1)==1,
(D[suppose,x]/. x→0)==-3,
(D[suppose,x]/. x→1)==9},{a,b,c,d}]
{{a→4,b→0,c→-3,d→0}}
```

因此，求得的多项式也是 4x^3-3x

9. **试构造一个四次埃尔米特插值多项式，使其满足下列函数表：**

x_i	0	1	2
$f(x_i)$	0	1	1
$f'(x_i)$	0	1	

解 调用麦卡的内部函数

```
InterpolatingPolynomial[{{0,{0,0}},{1,{1,1}},{2,1}},
x]//Factor
```

$$\frac{1}{4}(-3+x)^2 x^2$$

10. **已知函数 $y=f(x)$的函数表如下：**

x	0	1	4	5
$f(x)$	0	−2	−8	−4

在区间$[0,5]$**上求满足边界条件** $s'(0)=\frac{5}{2}$，$s'(5)=\frac{19}{4}$ **的三次样条插值函数** $s(x)$**，并分别计算** $s(x)$ **在** $x=0.5,3,5$ **处的值.**

解　根据已知，定义函数表

data$=\{\{0,0\},\{1,-2\},\{4,-8\},\{5,-4\}\}$;

d$=$**6Map[diffNewton, Partition[data, 3, 1]]**

$\{0,9\}$

h$=\{1,3,1\}$

$\{1,3,1\}$

$\lambda=\text{Table}\left[\frac{h[[i+1]]}{h[[i]]+h[[i+1]]},\{i,1,2\}\right]$

$\left\{\frac{3}{4},\frac{1}{4}\right\}$

$\mu=1-\lambda$

$\left\{\frac{1}{4},\frac{3}{4}\right\}$

$\text{d0}=\frac{6}{h[[1]]}\left(\text{diffNewton}[\{\text{data}[[2]],\text{data}[[1]]\}]-\frac{5}{2}\right)$

-27

$\text{dn}=\frac{6}{h[[3]]}\left(\frac{19}{4}-\text{diffNewton}[\{\text{data}[[4]],\text{data}[[3]]\}]\right)$

$\frac{9}{2}$

构造矩阵 M

$\text{m}=\left\{\{2,1,,0,0\},\left\{\frac{1}{4},2,\frac{3}{4},0\right\},\left\{0,\frac{3}{4},2,\frac{1}{4}\right\},\{0,0,1,2\}\right\}$;

myD=Prepend[Append[d,dn],d0]

$\left\{-27,0,9,\frac{9}{2}\right\}$

M=LinearSolve[m,myD]

$\left\{-\frac{27}{2},0,\frac{9}{2},0\right\}$

可确定三次样条插值函数 s(t)

s=

$$\text{Table}\Big[\frac{(\text{data}_{[[j+1,1]]}-x)^3}{6h_{[[j]]}}M_{[[j]]}+\frac{(x-\text{data}_{[[j,1]]})^3}{6h_{[[j]]}}M_{[[j+1]]}$$

$$+\left(\text{data}_{[[j,2]]}-\frac{M_{[[j]]}h_{[[j]]}^2}{6}\right)\frac{\text{data}_{[[j+1,1]]}-x}{h_{[[j]]}}$$

$$+\left(\text{data}_{[[j+1,2]]}-\frac{M_{[[j+1]]}h_{[[j]]}^2}{6}\right)\frac{x-\text{data}_{[[j,1]]}}{h_{[[j]]}},\Big]//$$

{j,1,3}

Expand

$$\left\{\frac{5x}{2}-\frac{27x^2}{4}+\frac{9x^3}{4},2-\frac{7x}{2}-\frac{3x^2}{4}+\frac{x^3}{4},,6-\frac{103x}{2}+\frac{45x^2}{4}-\frac{3x^3}{4}\right\}$$

对这三个分段函数作图(**题图** 2),**探讨曲线是否连接光滑**

Block[{ $DisplayFunction=Identity},

p1=Plot[s[[1]],{x,0,1}];

p2=Plot[s[[2]],{x,1,4}];

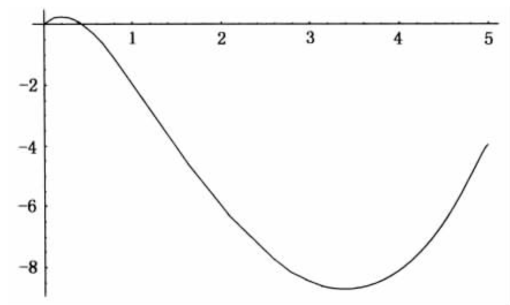
p3=Plot[s[[3]],{x,4,5}];];

Show[p1,p2,p3]

函数在三个点处的值

s[[1]]/. x→0. 5

−0. 156 25



题图 2

s[[2]]/. x→3

$-\frac{17}{2}$

s[[3]]/. x→5

-4

注:LinearSolve 是麦卡的内部函数,见“三”. 最后构造三次样条插值函数 s(t)的公式是按照书中的三弯矩法得到的.

13. 求函数 $f(x)=\sqrt{1+x^2}$ 在区间[0,1]上的一次最佳一致逼近多项式和一次最佳平方逼近多项式.

解 先求一次最佳一致逼近多项式

f[x_]=$\sqrt{1+x^2}$;

Simplify[f''[x]>0,0<=x<=1]

True

满足两次导数大于零的条件

$a_1=\frac{f[1]-f[0]}{1-0}$//N

0.414214

Solve[f'[x]==a_1,x]

{{x→0.45509}}

x_2=x/. First[%]

0.45509

$p1[x_]=\frac{f[0]+f[x_2]}{2}+a_1\left(x-\frac{0+x_2}{2}\right)$//N

1.04934+0.414214(−0.227545+x)

Expand[%]

0.95509+0.414214x

再求一次最佳平方逼近多项式

span={1,x}

{1,x}

m=Table[innerProduct[span[[i]],span[[j]],x,0,1],{i,1,2},{j,1,2}]

$\left\{\left\{1,\frac{1}{2}\right\},\left\{\frac{1}{2},\frac{1}{3}\right\}\right\}$

b=Table[innerProduct[span[[i]],f[x],x,0,1],{i,1,2}]

$\left\{\frac{1}{2}(\sqrt{2}+\text{ArcSinh}[1]),-\frac{1}{3}+\frac{2\sqrt{2}}{3}\right\}$

LinearSolve[m,b]//N

{0.93432,0.426947}

p[x _]=%[[1]]+%[[2]]x

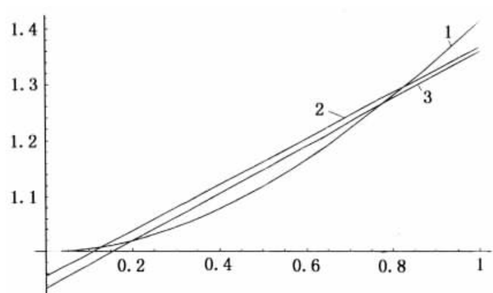
0.93432+0.426947x

Plot[{f[x],p1[x],p[x]},{x,0,1},PlotStyle→{{RGBColor[1,0,0]},{RGBColor[0,1,0]},RGBColor[0,0,1]}}]

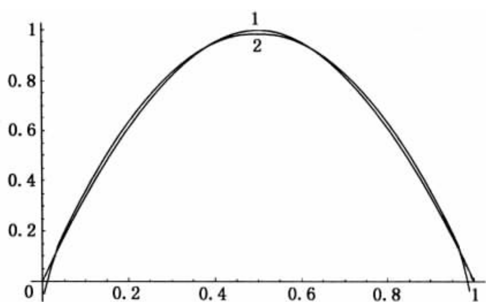
题图3中,曲线1是原函数,曲线2是一次最佳一致逼近多项式,曲线3是一次最佳平方逼近多项式.

14. **求函数** $f(x)=\text{Sin}\pi x$ **在**[0,1]**上的二次最佳平方逼近多项式.**

解　定义函数(**题图** 4)



题图 3



题图 4

```
f[x_]=Sin[πx];
span={1,x,x^2};
m=Table[innerProduct[span[[i]],
span[[j]],x,0,1],{i,1,3},{j,1,3}]
```

$$\left\{\left\{1,\frac{1}{2},\frac{1}{3}\right\},\left\{\frac{1}{2},\frac{1}{3},\frac{1}{4}\right\},\left\{\frac{1}{3},\frac{1}{4},\frac{1}{5}\right\}\right\}$$

```
b=Table[innerProduct[span[[i]],f[x],x,0,1],{i,1,3}]
```

$$\left\{\frac{2}{\pi},\frac{1}{\pi},-\frac{4}{\pi^3}+\frac{1}{\pi}\right\}$$

```
LinearSolve[m,b]//N
```

{-0.0504655,4.12251,-4.12251}

p[x _]=%[[1]]+%[[2]]x+%[[3]]x²

-0.0504655+4.12251x-4.12251x²

Plot[{f[x],p[x]},{x,0,1},

PlotStyle→{{RGBColor[1,0,0]},{RGBColor[0,1,0]}}]

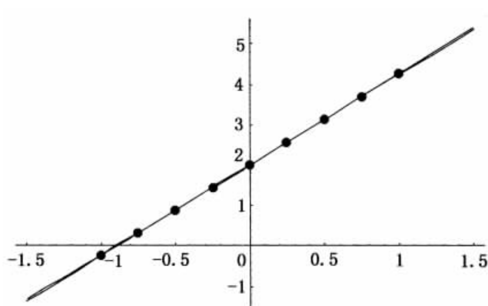
15. **给出数据**

x	-1.00	-0.75	-0.50	-0.25	0
y	-0.2209	0.3295	0.8826	1.4392	2.0003

x	0.25	0.50	0.75	1.00
y	2.5645	3.1334	3.7061	4.2836

希望用一次、二次和三次多项式,用最小二乘法拟合这些数据.

解　先定义数据,并将各个数据点用麦卡的内部函数 List-Plot **作图(题图** 5).



题图 5

data={{-1.00,-0.2209},{-0.75,0.3295},{-0.50,0.8826},{-0.25,1.4392},{0.,2.0003},{0.25,2.5645},{0.50,3.1334},{0.75,3.7061},{1.00,4.2836}};

gp=ListPlot[data,PlotStyle→PointSize[0.02],

DisplayFunction→Identity]

-Graphics-

分别用麦卡的内部函数 Fit **求取一次、二次和三次拟合多项式**

f1=Fit[data,{1,x},x]

2.01314+2.25165x

f2=Fit[data,{1,x,x^2},x]

2.0001+2.25165x+0.0313056x^2

f3=Fit[data,{1,x,x^2,x^3},x]

2.0001+2.25011x+0.0313056x^2+0.00208485x^3

Plot[{f1,f2,f3},{x,-1.5,1.5},

PlotStyle→{RGBColor[1,0,0],RGBColor[0,1,0],

RGBColor[0,0,1]},DisplayFunction→Identity]

-Graphics-

Show[gp,%,DisplayFunction→ $ DisplayFunction]

可以根据公式,计算二次拟合多项式的系数

Table[$\sum_{j=1}^{9}$data[[j]][[1]]$^{i+k}$,{i,0,2},{k,0,2}]

{{Indeterminate,0.,3.75},{0.,3.75,0.},{3.75,0.,
2.76563}}.

A=%/. Indeterminate→9

{{9,0.,3.75},{0.,3.75,0.},{3.75,0.,2.76563}}

Table[$\sum_{j=1}^{9}$data[[j]][[1]]^{i}data[[j]][[2]],{i,0,2}]

{Indeterminate,8.44367,7.58696}

b=%/. Indeterminate→Sum[data[[i]][[2]],{i,1,9}]

LinearSolve[A,b]

{2.0001,2.25165,0.0313056}

注：ListPlot 和 Fit 见“三”

16. **设有一发射源的发射强度公式为** $I=I_0e^{-\alpha t}$，**现测得** I **与** t **的一组数据如下：**

t_i	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I_i	3.16	2.38	1.75	1.34	1.00	0.74	0.56

试用最小二乘法根据上表确定参数 I_0 **和** α.

解

```
i={3.16,2.38,1.75,1.34,1.00,0.74,0.56};
lni=Log[i]
{1.15057,0.8671,0.559616,0.29267,0.,-0.301105,
-0.579818}
data=Transpose[{Range[0.2,0.8,0.1],lni}]
{{0.2,1.15057},{0.3,0.8671},{0.4,0.559616},
{0.5,0.29267},{0.6,0.},{0.7,-0.301105},{0.8,
-0.579818}}
Fit[data,{1,x},x]
1.72829-2.88829x
e^1.72829
5.63102
```

因此，公式中的 a=2.88829，I_0=5.63102

（Ⅱ）习题三

1. **分别用复合梯形公式和复合抛物线公式计算下列积分并比较结果：**(1) $\int_0^1 \frac{x}{4+x^2}dx$，($n=8$)；(2) $\int_1^9 \sqrt{x}dx$，($n=4$).

(1) **解**

```
f[x_]=x/(4+x^2);
rt=Map[f,Range[0,1,1./8]]
{0,0.0311284,0.0615385,0.090566,0.117647,
0.142349,0.164384,0.183607,0.2}
T=1/16(rt[[1]]+2Sum[rt[[i]],{i,2,8,}]+rt[[9]])
0.111402
rt2=Map[f,Range[0,1,1./16]]
{0,0.0156098,0.0311284,0.0464666,0.0615385,
0.0762631,0.090566,0.10438,0.117647,0.130317,
0.142349,0.153712,0.164384,0.17435,0.183607,
0.192154,0.2}
S=
1/48(rt2[[1]]+2Sum[rt2[[i]],{i,3,15,2}]
+4Sum[rt2[[i]],{i,2,16,2}]+rt2[[17]])
0.111572
NIntegrate[f[x],{x,0,1}]
0.111572
```

（2）解

```
f[x_]=√x;
rt=Map[f,Range[1,9,2.]]
{1,1.73205,2.23607,2.64575,3.}
T=(rt[[1]]+2Sum[rt[[i]],{i,2,4}]+rt[[5]])
17.2277
rt2=Map[f,Range[1,9,1.]]
{1,1.41421,1.73205,2.,2.23607,2.44949,2.64575,
2.82843,3.}
S=
```

```
1/3(rt2[[1]]+2Sum[rt2[[i]],{i,3,7,2}]
+4Sum[rt2[[i]],{i,2,8,2}]+rt2[[9]])
17.3321
NIntegrate[f[x],{x,1,9}]
17.3333
```

注：函数 Sum 是求和函数. NIntegrate 是麦卡内部函数，它通过数值积分法求函数积分的近似值.

4. 用代数精度定义直接验证抛物线求积公式具有 3 次代数精度.

解

mSimpson[a_,b_,f_]:= $\frac{b-a}{6}\left(f[a]+4f\left[\frac{a+b}{2}\right]+f[b]\right)$

Table[Integrate[x^i,{x,a,b}],{i,0,4}]

$\left\{-a+b,-\frac{a^2}{2}+\frac{b^2}{2},-\frac{a^3}{3}+\frac{b^3}{3},-\frac{a^4}{4}+\frac{b^4}{4},-\frac{a^5}{5}+\frac{b^5}{5}\right\}$

Table[mSimpson[a,b,#^i&],{i,0,4}]//Expand

$\left\{-a+b,-\frac{a^2}{2}+\frac{b^2}{2},-\frac{a^3}{3}+\frac{b^3}{3},-\frac{a^4}{4}+\frac{b^4}{4},\right.$

$\left.-\frac{5a^5}{24}+\frac{a^4b}{24}-\frac{a^3b^2}{12}+\frac{a^2b^3}{12}-\frac{ab^4}{24}+\frac{5b^5}{24}\right\}$

比较以上两个列表可得，直到三次多项式，求积公式还是精确成立的. 而对 $f(x)=x^4$ 求积公式不能精确成立. 因此说，抛物线公式有 3 次代数精度.

注：关于符号 # 和 & 的作用，可查阅麦卡关于纯函数(pure function)的说明.

5. 写出 $n=3$ 的牛顿-柯特斯公式，并求出其代数精度，利用此公式计算积分 $\int_0^1 \frac{1}{1+x^2}dx$ 的近似值.

解

mNC[a_,b_,f_]:=

$\frac{b-a}{8}\left(f[a]+3f\left[a+\frac{b-a}{3}\right]+3f\left[a+\frac{2(b-a)}{3}\right]+f[b]\right)$

Table[Integrate[x^i, {x,a,b}],{i,0,4}]

$\left\{-a+b,-\frac{a^2}{2}+\frac{b^2}{2},-\frac{a^3}{3}+\frac{b^3}{3},-\frac{a^4}{4}+\frac{b^4}{4},-\frac{a^5}{5}+\frac{b^5}{5}\right\}$

Table[mNC[a,b,#^i&],{i,0,4}]//Expand

$\left\{-a+b,-\frac{a^2}{2}+\frac{b^2}{2},-\frac{a^3}{3}+\frac{b^3}{3},-\frac{a^4}{4}+\frac{b^4}{4},\right.$

$\left.-\frac{11a^5}{54}+\frac{a^4b}{54}-\frac{a^3b^2}{27}+\frac{a^2b^3}{27}-\frac{ab^4}{54}+\frac{11b^5}{54}\right\}$

比较两个列表可见 n=3 **时,牛顿-柯特斯公式具有** 3 **次代数精度**

$f[x_]=\frac{1}{1+x^2}$;

mNC[0,1,f]//N

0.784615

7. **确定下列求积公式中的待定系数,使其代数精度尽量高,并指出其所具有的代数精度.**

(1) $\int_0^2 f(x)\mathrm{d}x\approx\omega_0 f(0)+\omega_1 f(1)+\omega_2 f(2)$;

(2) $\int_0^h f(x)\mathrm{d}x\approx\frac{h}{2}[f(0)+f(h)]+ah^2[f'(0)-f'(h)]$.

解 (1)

$\int_0^2 f(x)\mathrm{d}x\approx\omega_0 f(0)+\omega_1 f(1)+\omega_2 f(2)$

b=Table[Integrate[x^i, {x,0,2}],{i,0,2}]

$\left\{2,2,\frac{8}{3}\right\}$

m=Table[x^i,{i,1,2},{x,0,2}]

{{0,1,2},{0,1,4}}

m=Prepend[m,{1,1,1}]

$\{\{1,1,1\},\{0,1,2\},\{0,1,4\}\}$

ω=LinearSolve[m,b]

$\left\{\frac{1}{3},\frac{4}{3},\frac{1}{3}\right\}$

求出了三个待定系数的值,代回原求积公式检验其代数精度.

Table[$\mathbf{x^i}$,{i,2,4},{x,0,2}]

$\{\{0,1,4\},\{0,1,8\},\{0,1,16\}\}$

%.**ω**

$\left\{\frac{8}{3},4,\frac{20}{3}\right\}$

Table[Integrate[$\mathbf{x^i}$,{x,0,2}],{i,2,4}]

$\left\{\frac{8}{3},4,\frac{32}{5}\right\}$

从以上两个列表的比较可以看出,求积公式最高具有 3 次代数精度.

(4) $\int_0^h f(x)\mathrm{d}x \approx \frac{h}{2}[f(0)+f(h)]+ah^2[f'(0)-f'(h)]$

b=Table[Integrate[$\mathbf{x^i}$,{x,0,h}],{i,0,4}]

$\left\{h,\frac{h^2}{2},\frac{h^3}{3},\frac{h^4}{4},\frac{h^5}{5}\right\}$

mylist={1&,#&,#^{2}&,#^{3}&,#^{4}&,#^{5}&}

{**1**&,#**1**&,#**1^2**&,#**1^3**&,#**1^4**&,#**1^5**&}

b2=Table[$\mathbf{\frac{h}{2}}$(f[0]+f[h])+ah^2(f′[0]−f′[h])/.

f→mylist[[i]],{i,1,5}]

$\left\{h,\frac{h^2}{2},\frac{h^3}{2}-2ah^3,\frac{h^4}{2},-3ah^4,\frac{h^5}{2}-4ah^5\right\}$

Solve[$\mathbf{\frac{h^3}{3}==\frac{h^3}{2}-2ah^3}$,a]

$\left\{\left\{a \to \frac{1}{12}\right\}\right\}$

b2/. First[%]

$\left\{h, \frac{h^2}{2}, \frac{h^3}{3}, \frac{h^4}{4}, \frac{h^5}{6}\right\}$

上表和列表 b2 **比较,可知原求积公式具有** 3 **次代数精度**.

8. **用龙贝格求积算法,计算积分** $I = \int_0^{\pi} e^x \cos x \mathrm{d}x$.

解

f[x _]=e^{x}Cos[x];

Integrate[f[x],{x,0,π}]//N

−12.0703

mrTrp[1,f _,{a _,b _}]:=$\frac{b-a}{2}$(f[a]+f[b])

mrTrp[n _,f _,{a _,b _}]:=

myTrp$\left[\frac{n}{2}, f, \left\{a, \frac{b+a}{2}\right\}\right]$+myTrp$\left[\frac{n}{2}, f, \left\{\frac{b+a}{2}, b\right\}\right]$

tlist=Table[myTrp[2^{i},f,{0,π}]//N,{i,0,6}]

{−34.7785,−17.3893,−13.336,−12.3822,

−12.148,−12.0897,−12.0752}

myaccelerate[{a _,b _},k _ Integer]:=$\frac{4^k}{4^k-1}$b−$\frac{1}{4^k-1}$a

slist=Map[myaccelerate[#,1]&,Partition[tlist,2,1]]

{−11.5928,−11.9849,−12.0642,−12.07,

−12.0703,−12.0703}

clist=Map[myaccelerate[#,2]&,Partition[slist,2,1]]

{−12.0111,−12.0695,−12.0703,−12.0703,

−12.0703}

rlist=Map[myaccelerate[#,2]&,Partition[clist,2,1]]

{−12.0734,−12.0704,−12.0703,

-12.0703}

TableForm[{slist,clist,rlist}]

-11.5928	-11.9849	-12.0642	-12.07
-12.0703	-12.0703	-12.0111	-12.0695
-12.0703	-12.0703	-12.0703	-12.0734
-12.0704	-12.0703	-12.0703	

可见经过加速公式后，求得的结果-12.0703 和麦卡内部函数 NIntegrate 得出的结果很相近.

注：读者可用 NumberForm 函数使结果显示更多的有效位数.

9. 试用下述算法计算积分

$$I=\int_1^3 \frac{1}{y}\mathrm{d}y$$

(1) 利用龙贝格求积公式（计算到 R_1 为止）；

(2) 利用三点及五点高斯求积公式计算；

(3) 将积分区间分成四等分，在每一段用高点高斯求积公式，然后累加得 I 的值.（I 之精确值是 1.098612).

解 (1) 龙贝格求积公式

$$\mathbf{f[y_]=\frac{1}{y};}$$

tlist=Table[myTrp[2^i^,f,{1,3}]//N,{i,0,3}]

{1.33333,1.16667,1.11667,1.10321}

slist=Map[myaccelerate[#,1]&,Partition[tlist,2,1]]

{1.11111,1.1,1.09873}

clist=Map[myaccelerate[#,2]&,Partition[slist,2,1]]

{1.09926,1.09864}

rlist=Map[myaccelerate[#,2]&,Partition[clist,2,1]]

{1.0986}

(2) 高斯法

```
y[x_]=x^-1;
y[t_]=y[x]/.x->t+2
```

$\frac{1}{2+t}$

三点高斯求积公式

$\text{threepoints}=\frac{5}{9}y\left[\frac{-\sqrt{15}}{5}\right]+\frac{8}{9}y[0]+\frac{5}{9}y\left[\frac{\sqrt{15}}{5}\right]//N$

1.09804

五点高斯求积公式

```
fivepoints=
0.2369269y[0.9061798]+0.2369269y[-0.9061798]+
0.4786287y[0.5384693]+0.4786287y[-0.5384693]+
0.5688889y[0]
```

1.09861

(3) **分成四等分,每段用两点高斯求积公式**

```
Table[1+i(3-1)/4,{i,0,4}]
```

$\left\{1,\frac{3}{2},2,\frac{5}{2},3\right\}$

$\text{functionlist}=\left\{\frac{4}{5+\#}\&,\frac{4}{7+\#}\&,\frac{4}{9+\#}\&,\frac{4}{11+\#}\&\right\}$

$\left\{\frac{4}{5+\#1}\&,\frac{4}{7+\#1}\&,\frac{4}{9+\#1}\&,\frac{4}{11+\#1}\&\right\}$

$\text{Sum}\left[\frac{1}{4}\left(\text{functionlist}[[i]]\left[\frac{-1}{\sqrt{3}}\right]\right.\right.$

$\left.\left.+\text{functionlist}[[i]]\left[\frac{-1}{\sqrt{3}}\right],\{i,4\}\right]\right]//N$

1.09854

用麦卡的内部函数来求这个定积分

$$\int_1^3 \mathbf{y}^{-1}\mathbf{dy}//\mathbf{N}$$

1.09861

注:上面的定积分的符号可以直接在麦卡中输入,它完全等价于 NIntegrate[1/y,{y,1,3}]. 类似的,3 的平方根,可以用 Sqrt[3]输入,也可以输入根式符号. 麦卡有强大的数学符号和公式的输入及编辑功能,读者可试着从麦卡 3.0 或 4.0 系统的 File 菜单下的 Paletes 子菜单下找到相应的输入模板.

10. 已知函数 $y=f(x)$的数值表

x	1.0	1.1	1.2	1.3	1.4
$f(x)$	0.2500	0.2668	0.2066	0.1890	0.1736

用三点数值微分公式求 $f'(1.0)$,$f'(1.1)$和 $f'(1.2)$.

解 data={0.25,0.2268,0.2066};

$$\mathbf{r1}=\frac{\mathbf{1}}{\mathbf{0.2}}(-\mathbf{3data}[[\mathbf{1}]]+\mathbf{4data}[[\mathbf{2}]]-\mathbf{data}[[\mathbf{3}]]);$$

$$\mathbf{r2}=\frac{\mathbf{1}}{\mathbf{0.2}}(-\mathbf{data}[[\mathbf{1}]]+\mathbf{data}[[\mathbf{3}]]);$$

$$\mathbf{r3}=\frac{\mathbf{1}}{\mathbf{0.2}}(\mathbf{data}[[\mathbf{1}]]-\mathbf{4data}[[\mathbf{2}]]+\mathbf{3data}[[\mathbf{3}]]);$$

{**r1**,**r2**,**r3**}

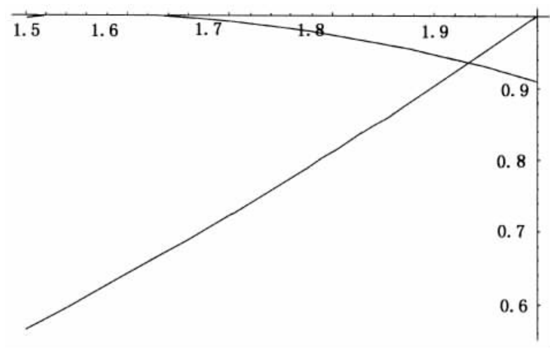
{-0.247,-0.217,-0.187}

(Ⅲ) 习题四

1. 用二分法求方程 $f(x)=\sin x-\left(\frac{x}{2}\right)^2=0$ 在区间[1.5,2]内的实根的近似值,并指出其误差.

解 见题图 6.

$$\mathbf{Plot}\left[\left\{\mathbf{Sin}[\mathbf{x}],\left(\frac{\mathbf{x}}{\mathbf{2}}\right)^{\mathbf{2}}\right\},\{\mathbf{x},\mathbf{1.5},\mathbf{2}\}\right]$$



题图 6

```
f[x_]=Sin[x]-(x/2)^2;
half[f_,{a_,b_}]:=
If[f[a]f[(a+b)/2]<0,{a,(a+b)/2},{(a+b)/2,b}]
NestList[half[f,#]&,{1.5,2},6]
{{1.5,2},{1.75,2},{1.875,2},{1.875,1.9375},
{1.90625,1.9375},{1.92188,1.9375},
{1.92969,1.9375}}
```

以上是经过 6 次迭代的结果,其误差小于 1/2^6.

注:NestList 见附录.

3. 用牛顿法求方程 $f(x)=x-\cos x=0$ 在 $x_0=1$ 附近的实根.

解

```
f[x_]=x-Cos[x];
g[x_]=x-f[x]/f'[x]
x-(x-Cos[x])/(1+Sin[x])
```

FixedPoint[g,1.]

0.739085

注:FixedPoint 见“三”.

4. **用牛顿法解方程 $x^3-a=0$,导出求立方根 $\sqrt[3]{a}$ 的近似公式.**

解

f[x_]=x³-a;

g[x_]=x-f[x]/f′[x]

$x-\frac{-a+x^3}{3x^2}$

5. **用弦截法求方程 $x^3-3x-1=0$ 在 $x_0=2$ 附近的实根.**

解

```
f[x_]=x^3-3x-1;
g[x_List]:=
Module[{t,xn=Take[x,-1][[1]],xs=Take[x,-2][[1]]},
t=xn-f[xn]/(f[xn]-f[xs])(xn-xs);Append[x,t]]
NestList[g,{1.9,2.0},5]
```

{{1.9,2.}{1.9,2.,1.88109},{1.9,2.,1.88109,1.87953},
{1.9,2.,1.88109,1.87953,1.87939},
{1.9,2.,1.88109,1.87953,1.87939,1.87939},
{1.9,2.,1.88109,1.87953,1.87939,1.87939,1.87939}}

经过五次迭代,结果为 1.87939.

(Ⅳ)习题五

3. **试用矩阵 A 的直接三角分解:$A=LU$ 求解方程组:**

(1) $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix}$;

(2) $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 256 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 10 \\ 44 \\ 190 \end{bmatrix}$

解 (1)

$$\boldsymbol{a} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{2} & \mathbf{2} & \mathbf{3} \\ -\mathbf{1} & -\mathbf{3} & \mathbf{0} \end{pmatrix};$$

LUDecomposition[**a**]

{{{1,2,1},{−1,−1,1},{2,2,−1}},{1,3,2},1}

LUBackSubstitution[%,{**0,3,2**}]

{1,−1,1}

(2)

$$\boldsymbol{a} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 256 \end{pmatrix};$$

LUDecomposition[**a**]

{{{1,2,3,4},{1,2,6,12},{1,3,6,24},{1,7,6,24}},{1,2,3,4},1}

LUBackSubstitution[%,{**2,10,44,190**}]

{−1,1,−1,1}

6. **已知**

$$A=\begin{pmatrix}1&1&1&1\\1&5&3&3\\1&3&11&5\\1&3&5&19\end{pmatrix}$$

试作 A 的乔列斯基分解：$A=LL^{\mathrm{T}}$，其中 L 是对角元全为正的下三角阵.

解

```
<<LinearAlgebra\Cholesky\
```

$$\boldsymbol{m}=\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}\\\mathbf{1}&\mathbf{5}&\mathbf{3}&\mathbf{3}\\\mathbf{1}&\mathbf{3}&\mathbf{11}&\mathbf{5}\\\mathbf{1}&\mathbf{3}&\mathbf{5}&\mathbf{19}\end{pmatrix};$$

```
u=CholeskyDecomposition[m]
{{1,1,1,1},{0,2,1,1},{0,0,3,1},{0,0,0,4}}
MatrixForm[u]
```

$$\begin{pmatrix}1&1&1&1\\0&2&1&1\\0&0&3&1\\0&0&0&4\end{pmatrix}$$

```
MatrixForm[Transpose[u].u]
```

$$\begin{pmatrix}1&1&1&1\\1&5&3&3\\1&3&11&5\\1&3&5&19\end{pmatrix}$$

9. **已知线性方程组** $\begin{cases}7x_1+10x_2=1\\5x_1+7x_2=0.7\end{cases}$

(1) **试求系数矩阵 A 的条件数** $\mathrm{Cond}_{\infty}(\boldsymbol{A})$；

(2) **若右端向量有扰动** $\delta\boldsymbol{b}=(0.01,-0.01)^{\mathrm{T}}$ **试估计解的相**

对误差.

解 （1）求系数矩阵 A 的条件数

```
<<LinearAlgebra\MatrixManipulation\
a={{7.,10},{5,7}};
MatrixConditionNumber[a]
```

289.

（2）对右端向量的扰动，估计相对误差

```
b={1,0.7};deltab={0.01,-0.01};
```

$$289\ \frac{\text{VectorNorm[deltab]}}{\text{VectorNorm[b]}}$$

2.89

10. 分别用雅可比迭代法与赛德尔迭代法求解方程组

$$\begin{cases} 20x_1+2x_2+\ \ 3x_3=24 \\ \ \ x_1+8x_2+\ \ \ x_3=12 \\ \ 2x_1-3x_2+15x_3=30 \end{cases}$$

取初值 $x^0=(0,0,0)^{\mathrm{T}}$，精确到小数后四位，并要求分别写出其迭代法的分量形式和矩阵形式.

解 （1）雅可比迭代法

$$A=\begin{pmatrix} 20 & 2 & 3 \\ 1 & 8 & 1 \\ 2 & -3 & 15 \end{pmatrix};$$

$$B_J=\text{Inverse}\left[\begin{pmatrix} 20 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 15 \end{pmatrix}\right]\begin{pmatrix} 0 & -2 & -3 \\ -1 & 0 & -1 \\ -2 & 3 & 0 \end{pmatrix}$$

$$\left\{\left\{0,-\frac{1}{10},-\frac{3}{20}\right\},\left\{-\frac{1}{8},0,-\frac{1}{8}\right\},\left\{-\frac{2}{15},\frac{1}{5},0\right\}\right\}$$

$$g_J=\text{Inverse}\left[\begin{pmatrix} 20 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 15 \end{pmatrix}\right].\{24,12,30\}$$

$\left\{\frac{6}{5},\frac{3}{2},2\right\}$

f[x _]：=$\boldsymbol{B}_{\mathrm{J}}$ · x+$\boldsymbol{g}_{\mathrm{J}}$

FixedPoint[f,{0. ,0. ,0. }]

{0.767354,1.13841,2.12537}

***A*. %**

{24. ,12. ,30. }

（2）**赛德尔迭代法**

$$\boldsymbol{U}=\begin{pmatrix}\mathbf{0} & \mathbf{-2} & \mathbf{-3}\\ \mathbf{0} & \mathbf{0} & \mathbf{-1}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix};$$

$$\boldsymbol{B}_{\mathrm{S}}=\textbf{Inverse}\left[\begin{pmatrix}\mathbf{20} & \mathbf{0} & \mathbf{0}\\ \mathbf{1} & \mathbf{8} & \mathbf{0}\\ \mathbf{2} & \mathbf{-3} & \mathbf{15}\end{pmatrix}\right].\boldsymbol{U}$$

$$\left\{\left\{0,-\frac{1}{10},-\frac{3}{20}\right\},\left\{0,\frac{1}{80},-\frac{17}{160}\right\},\left\{0,\frac{19}{1200},-\frac{1}{800}\right\}\right\}$$

$$\boldsymbol{g}_{\mathrm{S}}=\textbf{Inverse}\left[\begin{pmatrix}\mathbf{20} & \mathbf{0} & \mathbf{0}\\ \mathbf{1} & \mathbf{8} & \mathbf{0}\\ \mathbf{2} & \mathbf{-3} & \mathbf{15}\end{pmatrix}\right].\{\mathbf{24},\mathbf{12},\mathbf{30}\}$$

$\left\{\frac{6}{5},\frac{27}{20},\frac{211}{100}\right\}$

f[x _]：=$\boldsymbol{B}_{\mathrm{S}}$. x+$\boldsymbol{g}_{\mathrm{S}}$

FixedPoint[f,{0. ,0. ,0. }]

{0.767354,1.13841,2.12537}

***A*. %**

{24. ,12. ,30. }

11. **对方程组**$\begin{cases} x_1+2x_2-2x_3=1\\ x_1+x_2+x_3=2\\ 2x_1+2x_2+x_3=3\end{cases}$**分别讨论用雅可比迭代法**

与赛德尔迭代法求解的收敛性.

解

$$A=\begin{pmatrix}1 & 2 & -2\\ 1 & 1 & 1\\ 2 & 2 & 1\end{pmatrix};$$

$$U=\begin{pmatrix}0 & -2 & 2\\ 0 & 0 & -1\\ 0 & 0 & 0\end{pmatrix};$$

$$B_J=\text{Inverse}\left[\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\right].\begin{pmatrix}0 & -2 & 2\\ -1 & 0 & -1\\ -2 & -2 & 0\end{pmatrix}$$

{{0,-2,2},{-1,0,-1},{-2,-2,0}}

Eigenvalues[B_J]//N

{0.,0.,0.}

$$B_S=\text{Inverse}\left[\begin{pmatrix}1 & 0 & 0\\ 1 & 1 & 0\\ 2 & 2 & 1\end{pmatrix}\right].U$$

{{0,-2,2},{0,2,-3},{0,0,2}}

Eigenvalues[B_S]

{0,2,2}

12. **设方程组 $AX=b$ 中**

$$A=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & 1 & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2} & 1\end{bmatrix}$$

分别讨论用 Jacobi **迭代法**,G-S **迭代法求解的收敛性.**

解

$$A=\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix};$$

Det[*A*]>0

True

所以,赛德尔迭代法收敛

myD=IdentityMatrix[3];

MatrixForm[2 myD−*A*]

$$\begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}$$

Det[2 myD−*A*]

0

所以,雅可比迭代法发散.

B=Inverse[myD].

$$\begin{bmatrix} 0 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix}$$

$$\left\{\left\{0,-\frac{1}{2},-\frac{1}{2}\right\},\left\{-\frac{1}{2},0,-\frac{1}{2}\right\},\left\{-\frac{1}{2},-\frac{1}{2},0\right\}\right\}$$

Eigenvalues[B]//N

{－1. ,0.5,0.5}

14. **用逐次超松弛迭代法解方程组（取** $\omega=0.9, x^0=(0,0,0)^T$）

$$\begin{cases} 5x_1+2x_2+x_3=-12 \\ -x_1+4x_2+2x_3=20 \\ 2x_1-3x_2+10x_3=3 \end{cases}$$

精确到小数后四位.

解

$$\boldsymbol{U}=\begin{pmatrix} 0 & -2 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix};\mathbf{myD}=\begin{pmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 10 \end{pmatrix};$$

$$\boldsymbol{L}=\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ -2 & 3 & 0 \end{pmatrix};$$

B_ω＝Inverse[my*D*－0.9*L*].（0.1my*D*＋0.9*U*）

{{0.1,－0.36,－0.18},{0.0225,0.019,－0.4905},{－0.011925,0.06993,－0.000035}}

g_ω＝0.9Inverse[my*D*－0.9*L*].{－12,20,3}

{－2.16,4.014,1.74258}

f[*x* _]：＝B_ω.*x*＋g_ω

FixedPoint[f,{0. ,0. ,0. }]

{－4. ,3. ,2. }

（Ⅴ）习题六

1. **用欧拉法求解初值问题**

$$\begin{cases} y'=-y+x+1 \quad x\in[0,1] \\ y(0)=1 \end{cases}$$

$h=0.1$

解　先在麦卡中定义 $f(x,y)$

```
f[x_,y_]=-y+x+1
1+x-y
```

而后定义求解的起始点和步长

```
{x0,y0}={0,1};h=0.1;
```

计算的终点 x=1

```
xn=1
```

计算可得，一共需要计算 10 **个点上的值**

```
NumSteps=Ceiling[(xn-x0)/h]
10
```

根据欧拉公式，定义迭代计算的函数

```
euler[{x_,y_}]:={x+h,y+h*f[x,y]}
```

现在，就可以用麦卡的内部函数 NestList **进行迭代计算了**

```
result=NestList[euler,{x0,y0},NumSteps]
{{0,1},{0.1,1},{0.2,1.01},{0.3,1.029},
{0.4,1.0561},{0.5,1.09049},{0.6,1.13144},
{0.7,1.1783},{0.8,1.23047},{0.9,1.28742},
{1.,1.34868}}
```

将计算结果画成图形，把十个数据点连起来

```
eulerMethod=ListPlot[result,PlotJoined→True,
PlotStyle→RGBColor[1,0,0],
DisplayFunction→Identity]
-Graphics-
```

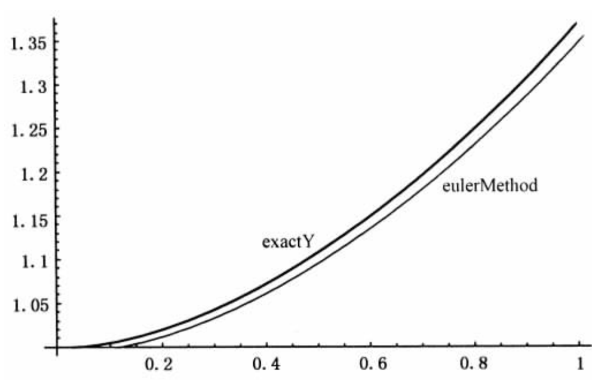
用麦卡的 Dsolve **内部函数可求得这道题的解析解**

```
DSolve[{y'[x]==f[x,y[x]],y[x0]==y0},y[x],x]
{{y[x]→e^-x(1+e^x x)}}
```

将它也画出函数图形，可以和欧拉法求解得到的近似值比较

```
exactY=Plot[e^-x(1+e^x x),{x,0,1.0},
DisplayFunction→Identity]
-Graphics-
Show[eulerMethod,exactY,
Graphics[{Text["exactY",{0.4,1.1}],
Text["eulerMethod",{0.9,1.2}]}],
DisplayFunction→ $DisplayFunction]
```

从题图 7 中可以看出，欧拉法还是可以近似地表示出原函数的图形.



题图 7

2. **标准四阶** R-K **方法求解初值问题**

$$\begin{cases} y'=y+x \quad x\in[0,1] \\ y(0)=1 \end{cases}$$

$h=0.2$

解 和上题一样，先定义

```
f[x_,y_]=x+y;
{x0,y0}={0,1};h=0.2;
```

定义标准的四阶 R-K 法迭代函数

```
RK[{x_,y_}]:=Module[{K1,K2,K3,K4},
K1=h*f[x,y];
K2=h*f[x+h/2,y+K1/2];
K3=h*f[x+h/2,y+K2/2];
K4=h*f[x+h,y+K3];
Return[{x+h,y+(1/6)(K1+2K2+2K3+K4)}]
RKt=NestList[RK,{x0,y0},5]
{{0,1},{0.2,1.2428},{0.4,1.58364},
{0.6,2.04421},{0.8,2.65104},{1.,3.4365}}
```

用 NestList 迭代求解，得出共 6 个点的函数值. 同样的，用麦卡的 Dsolve 可求得该问题的解析解.

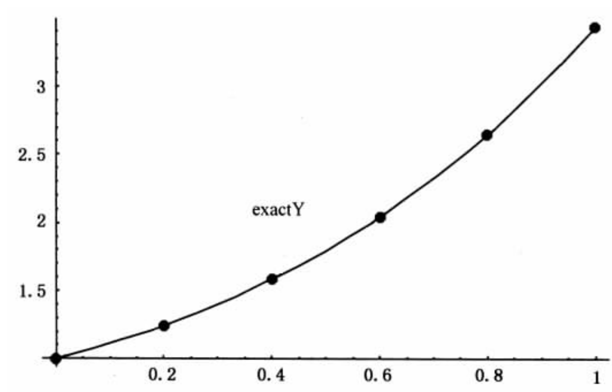
```
DSolve[{y'[x]==f[x,y[x]],y[0]==1},y[x],x]
{{y[x]→-1+2e^x-x}}
```

将结果分别作图，并显示在一起进行比较

```
RKMethod=
ListPlot[RKt,
PlotStyle→{PointSize[0.02],RGBColor[1,0,0]},
DisplayFunction→Identity]
-Graphics-
exactY=Plot[2e^x-1-x,{x,0,1.0},
DisplayFunction→Identity]
-Graphics-
Show [RKMethod,exactY,Graphics[Text["exactY",
```

```
        {0.4,2.1}]],
DisplayFunction→ $DisplayFunction]
```

可见,R-K 法得到的点基本上落在函数精确解的曲线上(题图 8).



题图 8

3. 验证公式

$y_{n+1}=y_n+h\psi(x_n,y_n,h)$是二阶 R-K 方法

其中

$$\psi(x_n,y_n,h)=\frac{1}{2}[f(x,y)+f(x+h,y+hf(x,y))]$$

解　将函数 y 作 Taylor 展开

```
Clear[f,y]
Normal[Series[y[x_{n+1}],{x_{n+1},x_n,2}]]/.(x_{n+1}-x_n)→h
```

$$y[x_n]+hy'[x_n]+\frac{1}{2}h^2y''[x_n]$$

求 y 的两次全微分

Dt[f[x,y[x]],x]

$y'[x]f^{(0,1)}[x,y[x]]+f^{(1,0)}[x,y[x]]$

根据函数 ψ 的定义,也将其展开 Taylor 级数,其中用到 $f(x,y)$ 的全微分

$$\boldsymbol{\psi[x_n, y_n, h]=}$$

$$\frac{\mathbf{1}}{\mathbf{2}}$$

$\{f[x_n,y[x_n]]+f[x_n,y[x_n]]+$

$hf[x_n,y[x_n]]\text{Derivative}[0,1][f][x_n,y[x_n]]+$

$h\ \text{Derivative}[1,0][f][x_n,y[x_n]])$

$\frac{1}{2}(2f[x_n,y[x_n]]+$

$hf[x_n,y[x_n]]f^{(0,1)}[x_n,y[x_n]]+hf^{(1,0)}[x_n,y[x_n]])$

由公式

$$\boldsymbol{y[x_{n+1}]=y[x_n]+h\psi[x_n,y_n,h]}$$

$y[x_n]+\frac{1}{2}h(2f[x_n,y[x_n]]+$

$hf[x_n,y[x_n]]f^{(0,1)}[x_n,y[x_n]]+hf^{(1,0)}[x_n,y[x_n]])$

Collect[%,h]

$hf[x_n,y[x_n]]+y[x_n]+$

$h^2\left(\frac{1}{2}f[x_n,y[x_n]]f^{(0,1)}[x_n,y[x_n]]+\frac{1}{2}f^{(1,0)}[x_n,y[x_n]]\right)$

可见,它和函数 y 的二阶 Taylor 展开相同,所以它是一个二阶的 R-K 方法.

4. **对初值问题,$\begin{cases} y'=-10y \\ y(x_0)=y_0 \end{cases}$由本章第 3 题给出的推算公式讨论绝对稳定性对步长 h 的限制.**

解　由第 3 题的结果,只要定义 $f(x,y)$

```
f[x_,y_]=-10y
```
$y[x_{n+1}]$

$y[x_n]+\frac{1}{2}h(-20y[x_n]+100hy[x_n])$

```
Coefficient[%,y[x_n]]
```
$1-10h+50h^2$

根据绝对稳定性的定义,可求得 h 的不等式,并使用麦卡求解

```
<<Algebra\InequalitySolve\
InequalitySolve[1-10h+50h^2<1,h]
```
$0<h<\frac{1}{5}$

6．**求系数 a,b,c,d 使得数值计算公式**

$$y_{n+1}=ay_{n-1}+h(by'_{n+1}+cy'_n+dy'_{n-1})$$

满足

$$y(x_{n+1})-y_{n+1}=O(h^5)$$

解　将各个 y 的离散形式用 Taylor 展开

```
y_{n+1}=Series[y[x_{n+1}],{x_{n+1},x_n,4}]//Normal
```
$y[x_n]+(-x_n+x_{1+n})y'[x_n]+\frac{1}{2}(-x_n+x_{1+n})^2y''[x_n]+$

$\frac{1}{6}(-x_n+x_{1+n})^3y^{(3)}[x_n]+\frac{1}{24}(-x_n+x_{1+n})^4y^{(4)}[x_n]$

```
y_{n+1}=y_{n+1}/.(x_{1+n}-x_n)→h
```
$y[x_n]+hy'[x_n]+\frac{1}{2}h^2y''[x_n]+\frac{1}{6}h^3y^{(3)}[x_n]+$

$\frac{1}{24}h^4y^{(4)}[x_n]$

```
y_{n+1}=y_{n+1}+O[h]^5;
```

求这些离散形式的导数：

$\mathbf{f_{n+1}=D[y_{n+1},x_n]}$

$y'[x_n]+hy''[x_n]+\frac{1}{2}h^2y^{(3)}[x_n]+\frac{1}{6}h^3y^{(4)}[x_n]+\frac{1}{24}h^4y^{(5)}[x_n]+O[h]^5$

$\mathbf{y_{n-1}=Series[y[x_{n-1}],\{x_{n-1},x_n,4\}]//Normal}$

$y[x_n]+(x_{-1+n}-x_n)y'[x_n]+\frac{1}{2}(x_{-1+n}-x_n)^2y''[x_n]+\frac{1}{6}(x_{-1+n}-x_n)^3y^{(3)}[x_n]+\frac{1}{24}(x_{-1+n}-x_n)^4y^{(4)}[x_n]$

$\mathbf{y_{n-1}=y_{n-1}/.(x_{n-1}-x_n)\to -h}$

$y[x_n]-hy'[x_n]+\frac{1}{2}h^2y''[x_n]-\frac{1}{6}h^3y^{(3)}[x_n]+\frac{1}{24}h^4y^{(4)}[x_n]$

$\mathbf{y_{n-1}=y_{n-1}+O[h]^5;}$

$\mathbf{f_{n-1}=D[y_{n-1},x_n]}$

$y'[x_n]-hy''[x_n]+\frac{1}{2}h^2y^{(3)}[x_n]-\frac{1}{6}h^3y^{(4)}[x_n]+\frac{1}{24}h^4y^{(5)}[x_n]+O[h]^5$

求解由待定系数 a,b,c,d **组成的方程**

$\mathbf{LogicalExpand[y_{n+1}=ay_{n-1}+h(bf_{n+1}+cy'[x_n]+df_{n-1})]}$

$-y[x_n]+ay[x_n]==0\&\&$

$-y'[x_n]-ay'[x_n]+by'[x_n]+cy'[x_n]+dy'[x_n]==0\&\&$

$-\frac{1}{2}y''[x_n]+\frac{1}{2}ay''[x_n]+by''[x_n]-dy''[x_n]==0\&\&$

$-\frac{1}{6}y^{(3)}[x_n]-\frac{1}{6}ay^{(3)}[x_n]+\frac{1}{2}by^{(3)}[x_n]+\frac{1}{2}dy^{(3)}[x_n]==0\&\&$

$-\frac{1}{24}y^{(4)}[x_n]+\frac{1}{24}ay^{(4)}[x_n]+\frac{1}{6}by^{(4)}[x_n]-\frac{1}{6}dy^{(4)}[x_n]$

==0

Solve[%,{**a**,**b**,**c**,**d**}]

$$\left\{\left\{c\to\frac{4}{3},b\to\frac{1}{3},d\to\frac{1}{3},a\to 1\right\}\right\}$$

3. 麦卡函数调用说明

麦卡部分内部函数调用规则及使用说明

▌ InterpolatingPolynomial

该函数是用来逼近插值多项式，调用规则 InterpolatingPolynomial[data,var]，**其中** data **是插值多项式在** n **个节点上满足的插值条件，插值条件** data **可以是多重的列表**{{x1,f1},{x2,f2},…}，var **则是得出的用牛顿插值多项式表示的多项式中的变量名，一般设为** x. **见习题二**(1)**中的例子. 此外，数表**{x1,f1}**还可变为**{x1,{f1,df1,ddf1,…}}，**也就是将该节点处的导数条件包括进去. 见习题二**(9)**中的例子.**

▌ Map

Map[f,expr]**是将函数作用于** expr **中的每个元素. 最常见的** expr **就是数列了. 例如**，data={1,2,3}，Map[Sin,data]**就会得出**{Sin[1],Sin[2],Sin[3]}. **当然，直接执行** Sin[data]**是一样的，这是由于函数** Sin **可对数列操作. 用** Attributes[Sin]**可以看到，它具有** Listable **这一属性. 这样的函数就可以对数列的各个元素进行运算，并且结果还是一个数列. 但是不是所有的函数都具有** Listable **的属性，这时就要借助** Map **函数了. 例如：**

Map[NumberForm[#,10]&,{Sin[1.0],Sin[2.0],Sin[3.0]}]**就是将数列中的三个** Sin **函数的值表示到** 10 **位有效数字. 这里，**# **和** & **两个符号的作用请查阅麦卡帮助中关于纯函数**(pure function)**的说明.**

Partition

Partition[list,n,d]，顾名思义是将一个列表分离成几个列表.例如：

Partition[{1,2,3,4,5,6},2]是将数表分成每段两个元素的列表，得{{1,2},{3,4},{5,6}}

Partition[{1,2,3,4,5,6},2,1]也是将数表分成每段两个元素的列表，不过有一个元素的重叠，得到

{{1,2},{2,3},{3,4},{4,5},{5,6}}

LinearSolve

LinearSolve[m,b]，求解线性方程 m. x==b

ListPlot

ListPlot[{{x1,y1},{x2,y2},…}]

将 x,y 坐标对应的各个点画在图上.该函数输出一个图形对象，也可以用一个变量来命名.例如：

mygraph=ListPlot[{1,3},{2,4},DisplayFunction→Identity]

产生一个名为 mygraph 的图形对象.这里，Display Function→Identity是使输出的图形暂时不显示，而只是以 mygraph 的名字保存起来.要想显示该图形，可以调用 Show[mygraph,DisplayFunction→$DisplayFunction]来实现.这有利于同时将几个图形显示在一起，而在此之前，单个的图形不需要显示出来.

Show

Show[graphics,options]，可以分别将几个二维或三维的图形对象同时显示在一幅图内.其中，options 选项可以包括许多显示图形时的设置.这样，可以实现图形间的比较和图形对象的组合.

▌Fit

Fit[data,Funs,vars],通过最小二乘法拟和出由 funs 中的基函数组成的多项式,而 vars 是指多项式中的变量名.

例如:

Fit[{{x1,f1},{x2,f2},…,}],{1,x,x^2},x]

就是通过一个 x 的二次多项式拟和数据.

▌NestList

NestList[f,expr,n]将函数 f 作用于表达式 expr 连续 n 次.例如:

NestList[Sin,0.1,5]相当于 Sin[Sin[Sin[Sin[Sin[1]]]]]

▌FixedPoint

FixedPoint[f,expr]和 NestList 相似,它也是将函数 f 连续作用于表达式 expr.不过它直到得出的结果的值不再变化才停止运算.该函数适于实现收敛的迭代运算.当然,它还有更说细的运算停止条件的控制,请参阅麦卡的在线帮助.

二、MATLAB 使用初步

1. MATLAB 简介

MATLAB 是英语 Matrix Laboratory 的缩写.它诞生于一个学习线性代数的教学软件.经过不断的完善和发展,如今它已是一个功能强大的、广泛应用于科学和工程计算的计算机软件系统.使 MATLAB 如此成功的原因之一是其具有的工具箱(Toolbox)这一软件包形式.这些工具箱都是针对某个工程或科学领域中的计算问题,邀请该领域的专家,编写出多个可供方便调用而且功能强

大的函数,并将这些函数集合在一个称为工具箱的软件包内.如信号处理工具箱、控制系统工具箱、神经网络工具箱等二三十个各种各样的工具箱.这些工具箱中的函数可以在 MATLAB 系统中被方便地调用,甚至其中不少是用 MATLAB 提供的编程语言写的,所以可以进行修改和扩展.因此说,MATLAB 的可扩展性和通用性是做得很好的,这也使得它被广泛地应用于工程和科学的各个领域.MATLAB 系统中的另一个很有特色的构件就是 Sinulink 系统,它可以对动态系统模型进行数字仿真,并且其图形化设计界面使得构造系统模型的工作非常直观和方便.

如今,MATLAB 系统的 6.0 版本已经发行,新增加了多个领域的工具箱,功能也不断地加强了.由于这里只是一个对于 MATLAB 系统的简介,我们并不打算涉及到各个领域工具箱的用法.并且对于数值计算基础课程中的编程和计算,只要使用 MATLAB 基本系统所提供的功能就足够了.好在一旦学会了 MATLAB 的基本用法,再去使用与你所感兴趣领域相关的工具箱,就很容易入门了.并且 MATLAB 提供了很详细的在线文档,往往是一个工具箱就有一本专门的手册进行介绍.这些手册可以随系统安装在计算机上,阅读它们是学习 MATLAB 使用的最好工具了.按照 MATLAB 在线手册的说法,MATLAB 可分为五个部分:

(1) MATLAB 编程语言

(2) MATLAB 集成系统

(3) MATLAB 图形系统

(4) MATLAB 的大量数学计算库函数

(5) MATLAB 软件接口函数

简单说明一下这五个部分的关系:首先你在 UNIX 或 WINDOWS 系统中安装完 MATLAB 系统后,一旦开始运行 MATLAB,出现的就是它的集成环境.你可以在其中输入命令,执行代码或调用软件包中的函数,打开编辑窗口编写 MATLAB 程序等工作.因此,MATLAB 给用户的外部界面就是这个集成系统.当

然，进入了这个集成系统，你就得用 MATLAB 的语言和它交流，而这就是 MATLAB 的编程语言．熟悉 UNIX 系统的读者可以发现，这套编程语言和 UNIX 中的一些脚本(script)语言的风格很相像．甚至 MATLAB 语言中也有 ls，who 之类的与 UNIX 系统命令同名的命令．就如脚本语言一样，MATLAB 语言中的命令可以一名一句地在集成环境中执行，它们也可以写成一个脚本，在 MATLAB 中称为 M 文件(M-file)，而后在集成环境中作为一段程序代码执行．MATLAB 也提供了一般编程语言所必需的流程控制的如 if，for，while 等语句，可以实现传统的 C 或 FORTRAN 语言的程序功能．而用 MATLAB 语言编写程序的优点之一就是 MATLAB 对于矩阵或向量的操作很方便，这也许要归因于它原来就是一个用来学习线性代数的软件吧．还有许多现成的 MATLAB 内部函数可供编程者调用，这就省去了不少自己写代码的麻烦．

以下就分为两个部分来简单介绍 MATLAB 基本系统的使用方法．最后的应用 MATLAB 的计算实例中只是列举了几个在数值计算基础课程中的习题通过 MATLAB 解答的例子，希望读者举一反三，能够借助于 MATLAB 学习数值计算课程中的一些算法和理论，并能够将其中的一些实现为 MATLAB 的程序．

(Ⅰ) MATLAB 中的向量与矩阵

由于 MATLAB 是一个解释执行的脚本语言系统，在其中定义变量不需要事先声明变量类型．

例如定义矩阵 $\boldsymbol{A}$，在 MATLAB 中输入

```
A=[1 2 3;4 5 6;7 8 9]
A=
     1    2    3
     4    5    6
     7    8    9
```

再运行命令 sum($\boldsymbol{A}$)

ans=

    12   15   18

可见结果是将矩阵 *A* 按列相加起来，要查看 sum **函数的用法，可输入**

```
help sum
    SUM Sum of elements.
    For vectors, SUM(X) is the sum of the elements of X.
    For matrices, SUM(X) is a row vector with the sum over
    each column. For N-D arrays, SUM(X) operates along
    the first non-singleton dimension.
    SUM (X,DIM) sums along the dimension DIM.
    Example: If X=[0 1 2
                   3 4 5]
    then sum (X,1) is [3 5 7] and sum (X,2) is [3
                                                 12];
    See also PROD, CUMSUM, DIFF.
```

对于 MATLAB **中的其他函数，都可用** help **命令在线的查询其用法. 从上面的这段用法说明中可以看到，要对矩阵 *A* 的行元素求和，只需调用** sum(***A***,2)**即可**

ans=

      6
     15
     24

要取 *A* 中的某个元素也很方便，例如输入 *A*(1,2)

ans=

     2

就得到矩阵 *A* 的第一行第二个元素了. 要取得 *A* 的对角线元

素，也有现成的函数可以调用

```
diag(A)
ans=
     1
     5
     9
```

试想这些功能，要用传统的 C 语言编程，也要花费不少功夫.

在定义数组或矩阵时，有一个运算符很有用，就是冒号：

例如，要产生一个 1～10 的数列，可以在 MATLAB 中输入

```
1:10
ans
   1   2   3   4   5   6   7   8   9   10
```

同样的，如果要取先前定义的矩阵 A 的第一列，可以输入

```
A(:,1)
ans=
     1
     4
     7
```

另外要说明的是，MATLAB 中二维数组，也就是矩阵是按列排的，这和 C 语言的习惯不同，而是沿袭了 FORTRAN 语言的风格. 从以下的调用可以看出，A 的第一个元素是 1，而第二个元素为 4，而不是 2.

```
A(1)
ans=
     1
A(2)
```

```
ans=
    4
```

再定义一个矩阵 *B*

```
B[1 0 -1;0 1 2; 0 0 4]
B=
    1   0   -1
    0   1   2
    0   0   4
```

两个同阶矩阵可以进行加、减、乘的运算.例如:

```
A+B
ans=
    2   2   2
    4   6   8
    7   8   13
A*B
ans=
    1   2   15
    4   5   30
    7   8   45
```

一般,矩阵运算中的求逆,行列式的值,特征值,特征向量等,都有内部函数可调用.

```
invB=inv(B)
invB=
    1.0000         0    0.2500
         0    1.0000   -0.5000
         0         0    0.2500
```

```
invB * B
ans=
      1   0   0
      0   1   0
      0   0   1
[V,D]=eig(B)
V=
       1.0000          0   -0.2673
            0     1.0000    0.5345
            0          0    0.8018
D=
      1   0   0
      0   1   0
      0   0   4
```

其中,V 中的无向量是特征向量,而 D 中对角线元素是对应的特征值.

MATLAB 中对矩阵操作的函数还有许多,在线手册《Using MATLAB》中专门有一个叫做“矩阵和线性代数”的章节对此进行了介绍.有兴趣的读者可参考其中的介绍.

(Ⅱ) MATLAB 的函数与脚本

从前面的介绍中我们已经看到许多 MATLAB 函数调用的例子了.读者可以发现,MATLAB 中的函数调用是很灵活的.例如前面的 sum 函数可以带一个参数,也可以带两个参数.更不同于传统计算机编程语言的是 MATLAB 的函数的输出值可以不止一个,例如,前面求矩阵特征值和特征向量的内部函数 eig 就可以分别输出特征值和特征向量到两个变量

```
[V,D]=eig(B)
V=
```

```
      1.0000         0   -0.2673
           0    1.0000    0.5345
           0         0    0.8018
D=
     1    0    0
     0    1    0
     0    0    4
```

以下我们就来说明用户怎样编写出如此功能强大的 MATLAB **函数.**

首先,通过使用 MATLAB **语言写的代码存成的文件称为** M **文件. 它当然是以**.m **作为文件后缀名. 可以用简单的文本编辑器来编写这个** M **文件,也可以使用** MATLAB **自带的代码编辑器. 例如,在编辑器中输入如下的代码:**

```
function c=mytest (a,b)
c=sqrt (a^2+b^2);
```

这段代码定义了一个函数,该函数所实现的功能很简单:将两个数的平方和开根号. 把这段代码保存为 mytest.m **文件,注意文件名一定要和函数名相同. 文件要存在** MATLAB **中设置的路径的范围内,这样,当要调用该函数时,**MATLAB **可自动找到它. 如果使用的是** MATLAB5.2 **及以后的版本,它在** MATLAB **的安装目录下提供了一个名为** Work **的子目录,可以将自定义的函数文件存放在这个子目录下. 现在就可以在** MATLAB **的集成环境中调用自定义的函数了. 输入:**

```
a=3;b=4;
c=mytest(a,b)
C=
     5
```

读者也许注意到了，前面定义函数和定义变量时使用了分号“;”，它表示不将计算结果显示出来，读者可以试着去掉函数定义中的那个分号，看看调用后的结果. M 文件不光可用来定义函数，还可以用来写脚本. 脚本也是让 MATLAB 做计算的代码，和函数不同的是，脚本不输出参数，也不要求输入参数，这也许有些类似于 DOS 系统中的批处理(.bat)文件吧.

下面是在线手册中的一个脚本的例子：

```
%An M-file script to produce "flower petal" plots
theta=-pi:0.01:pi;
rho(1,:)=2*sin(5*theta).^2;
rho(2,:)=cos(10*theta).^3;
rho(3,:)=sin(theta).^2;
rho(4,:)=5*cos(3.5*theta).^3;
for i=1:4
ploar(theta,rho(i,:))
pause
end
```

把它存为 petal.m 的文件，就可以在 MATLAB 集成环境中运行该脚本了.

再来看一个比较完整的 MATLAB 函数定义的例子：

```
function y=average(x)
% AVERAGE Mean of vector elements.
% AVERAGE(X), where X is a vector, is the mean of vector elements.
% Non-vector input results in an error.
[m,n]=size(x);
if(~((m==1)|(n==1))|(n==1&n==1))
    error('Input must be a vector')
```

```
end
y=sum(x)/length(x);% Actual computation
```

这是一个求向量的平均值的函数. 同样的,将其保存为 average.m 的文件. 调用一下看:

```
z=1:99;
average(z)
ans=
     50
```

代码的第一行中 function 是一个关键字,表示所定义的是一个函数. 其后分别是函数名和它的输入参数和输出参数.

第二行中带%开头的是注释语句,它们往往说明该函数的功能和用法. 在输入 help 命令查询函数用法时,系统就将这里的注释显示出来.

```
help average
    AERAGE Mean of vector elements.
    AVERAGE(X), where X is a vector, is the mean of vector
    elements.
    Non-vector input results in an error.
```

再接下来就是函数体了,代码

```
[m,n]=size(x);
if(~((m==1)|(n==1))|(m==1&n==1))
error('Input must be a vector')
end
```

是通过 if 语句检查输入的参数是否是一个一维的数组(包括行向量或列向量).

在保证输入的参数正确的情况下,代码

```
y=sum(x)/length(x);
```

进行求平均值的计算,并将结果赋给内部变量 y 返回.

函数定义时,MATLAB 提供两个变量 nargin 和 nargout,分别表示调用该函数时输入参数和输出参数的个数,知道了这些值,程序就可以针对不同的情况分别处理了.

例如

```
function c = testargl(a,b)
if (nargin==1)
c=a.^2;
elseif (nargin==2)
c=a+b;
end
```

表示如果输入参数为一个,就将其平方.如果为两个,就将两者求和.这里,变量 a 后面加了一点“.”表示平方的运算是对 a 中的每个元素进行的,这主要是由于输入的参数 a 可能是一个一维向量.而该函数可以对向量中的每个元素进行操作.MATLAB 之所以对向量和矩阵的操作功能强大,是因为它在定义内部函数时往往考虑到了对于向量或矩阵的运算的支持.因此,用户在写自定义函数时,也应该注意保持这方面的通用性.

MATLAB 作为一种编程的脚本语言,当然提供了控制程序流向的 if,switch,for,while 等语句.它们的用法和传统的编程语言中的相似,这里就不再说明了.

此外,MATLAB 函数的参数是传值型的.也就是说,输入的参数即使在函数中参与了运算,函数返回后它们的值是不会被改变的.你也不能像 C 语言那样将一个变量的内存地址用指针的形式传给函数.从而改变输入参数的值.好在 MATLAB 函数的输出形式可以很灵活,数据结构也很丰富,虽以满足编程的需要.

在 MATLAB 的函数里,也可以接受用户的输入,例如

```
help input
    INPUT Prompt for user input.
    R=INPUT ('How many apples') gives the user the prompt
    in the text string and then waits for input from the key-
    board.
    The input can be any MATLAB expression, which is evalu-
    ated, using the variables in the current works-pace, and the
    result returned in R. If the user presses the return key
    without entering anything, INPUT returns an empty ma-
    trix.
    R=INPUT ('What is your name','s')gives the prompt in
    the text string and waits for character string input. The
    typed input is not evaluated; the characters are simply re-
    turned as a MATLAB string.
    The text string for the prompt may contain one or more '\
    n'. The '\n' means skip to the beginning of the next line.
    This allows the prompt string to span several lines. To out-
    put just a '\' use '\\'.
    See also KEYBOARD.
```

可见 input **这个命令就可用来接受用户输入.**

```
n=input ('what is your age?')
```

会在集成环境中显示“what is your age**?’的提示,然后等待用户的输入,并且把输入的值赋给变量** n.

还有前面出现过的 pause **命令,它会使程序在执行过程中暂停,等待用户按键盘上的任意一个键,而后再继续.它也是程序和用户交互的很有用的一个命令.** MATLAB **编程语言的基本内容就是这些了.但是要编写出好的代码,有效地利用** MATLAB **系统的功能,熟悉其丰富的内部函数很重要.因为在** M **文件中和在集**

成环境中一样，可以方便地调用 MATLAB 的内部函数. 这样，用户可以不必重复编写 MATLAB 已有的程序代码，大大提高了开发效率. 另外，使用 MATLAB 功能强大的向量操作的功能，也可使程序代码得到简化. 例如，下面两段代码完成的是同样的工作.

```
i=0;
for t=0:0.01:10
i=i+1;
y(i)=sin(r);
end
```

使用向量的形式，就可避免使用 for 循环语句了，代码的执行速度也会提高.

```
t=0:.01:10;
y=sin(t);
```

更详细的介绍请查阅在线手册《Using MATLAB》中的"M-File Programming"一章.

2. 应用 MATLAB 的计算实例

例 1 用标准四阶 R-K 方法求解初值问题

$$\begin{cases} y'=y+x & x\in[0,1] \\ y(0)=1 \end{cases}$$

$h=0.01$

可以用 MATLAB 编程语言来定义 RungeKutta 函数.

```
function [xout,yout]=RungeKutta(Fcn,a,b,stepsize)
%RungeKutta Method
xout=a(1):stepsize:b(1);
yout=zerose(size(xout));
yout(1)=a(2);
```

```
for I=1:size(xout,2)-1,
yout(i+1)=RK(Fcn,xout(i),yout(i),stepsize);
end
finction yout=RK(Fcn,x,y,h)
k1=h*feval(Fcn,x,y);
k2=h*feval(Fcn,x+h/2,y+k1/2);
k3=h*feval(Fcn,x+h/2,y+k2/2);
k4=h*feval(Fcn,x+h,y+k3);
yout=y+(k1+2*k2+2*k3+k4)/6.0;
```

将以上代码存为 RungeKutta.m **文件，而后定义函数** f(x,y)

```
function c=Fxy(x,y)
c=x+y;
```

将其也存为一个 M **文件. 现在就可以在** MATLAB **集成环境中调用这些函数求解了.**

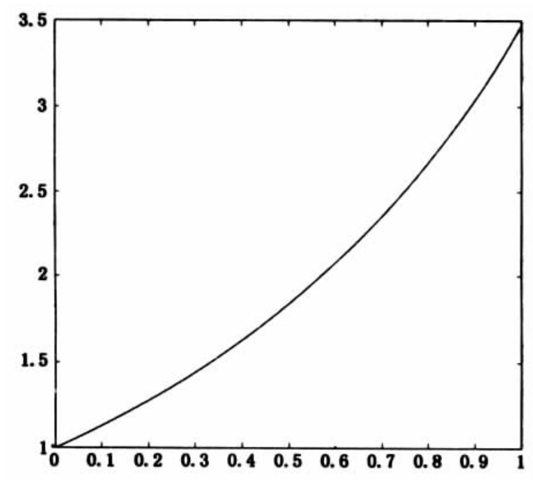
```
[x,y]=RungeKutta ('Fxy',[0,1],1,0.01);
plot(x,y)
```

例 2　用追赶法求解三对角线性代数方程组

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{4} \end{pmatrix}$$

先按照追赶法编写程序如下：

```
function x=TraceMethod(A,d)
%Solve linear equations using trace method
```



附图 6

```
b=diag(A);
c=diag(A,1);
a=diag(A,-1);
y=zeros(size(A,1),1);
l=zerops(size(b,1),1);
u=zeros(size(c,1),1);
m=a;
l(1)=b(1);
u(1)=c(1)/l(1);
y(1)=d(1)/l(1);
for i=2:size(A,1),
u(i-1)=c(i-1)/l(i-1);
l(i)=b(i)-m(i-1)*u(i-1);
y(i)=(d(i)-m(i-1)*y(i-1))/l(i);
end
```

```
x=zeros(size(A,1),1);
x(size(A,1))=y(size(A,1));
for i=size(A,1)-1:-1:1,
x(i)=y(i)-u(i)*xi+1);
end
```

将文件存为 TracMethod.m 就可以在集成环境中调用了.

```
A=  [2,-1,0,0;
    -1,2,-1,0;
    0,-1,2,-1;
      0,0,-1,2]
A=
  2  -1   0   0
 -1   2  -1   0
  0  -1   2  -1
  0   0  -1   2
TraceMehtod(A,[1;1/2;1/3;1/4])
aus=
      1.2833
      1.5667
      1.3500
      0.8000
```

这里的自定义函数都只是非常简单的示例.而一个完整的、容借性强的程序,往往首先要对输入参数进行检验,看是否符合求解条件.不符合条件的,就要显示出错信息,提示用户重新输入.读者如果查看 MATLAB 的内部函数的定义,就会发现 MATLAB 的内部函数在这方面的功能是很完备的.

以上的 Mathematica 代码是在 Mathematica 4.0 系统中实现的.MATLAB 的代码是在 MATLAB 5.0 系统中实现的.所有的

运算结果均是在 Pentium Ⅱ 的微机上运行得到的. 需要再次指出的是,在数值计算基础课程的学习中,通过使用数学软件进行数学实验是必要的. 但数学软件主要还是作为一个学习和解决问题的工具,而对数值计算方法本身思想的了解和掌握才是理工科大学生数学素质的体现.

实习题四

(用 Matlab 软件)

1. 用形如 $ae^{x}+b\sin(x)+c\ln(x)+d\cos(x)$ 的函数在最小二乘的意义下拟合数据表:

x	0.25	0.5	0.75	1.00	1.25	1.5	1.75	2.00	2.25	2.50
y	1.284	1.648	2.117	2.718	3.427	2.798	3.534	4.456	5.465	5.894

2. 函数 $y=f(x)$ 由下列数据给出:

k	0	1	2	3	4	5	6	7	8
x_k	−1.00	−0.75	−0.50	−0.25	0.00	0.25	0.5	0.75	1.00
y_k	0.221	0.329	0.883	1.439	2.003	2.564	3.133	3.706	4.284

试用 $(x_3,y_3),(x_4,y_4),(x_5,y_5),(x_6,y_6)$ 构造三次多项式,求出 $y=f(x)$ 在 $x=0.13$ 处的近似值.

3. 计算积分 $I=\int_{0.1}^{0.5}\frac{\arctan(x)}{x^2}\mathrm{d}x$.

4. 求方程 $x^9-522+e^x=0$ 在 1.9 附近的根.

5. 取 $h=0.01$,求解下列初值问题:

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=\sqrt{x^2+y^2} & 0.0\leqslant x\leqslant 1.0\\ y(0)=-1.\end{cases}$$

要求写出最后 8 个点上的函数值.

6．讨论用 Jacobi 迭代法求解方程组 $\boldsymbol{AX}=\boldsymbol{b}$ 的收敛性，其中，$\boldsymbol{A}$ 为如下的 100 阶的方阵，$\boldsymbol{b}$ 为如下的 100 维的列向量. 并要求写出简单的理由.

$$\boldsymbol{A}=\begin{bmatrix} \frac{1}{1}+\frac{1}{2} & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3}+\frac{1}{2} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1}+\frac{1}{2} \end{bmatrix},$$

$$\boldsymbol{b}=\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

7．考虑一种特殊的对角线元素不为 0 的双对角线性代数方程（以 $N=7$ 为例）

$$\begin{bmatrix} d_1 & & & & & & \\ a_1 & d_2 & & & & & \\ & a_2 & d_3 & & & & \\ & & a_3 & d_4 & a_4 & & \\ & & & & d_5 & a_5 & \\ & & & & & d_6 & a_6 \\ & & & & & & d_7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}$$

用 MATLAB 语句写出解上述一般的 N（奇数）阶方程组的 m 函数 $x=\text{bidiag}(N,a,d,b)$，其中，x,a,d,b 均为长度为 N 的数组.

8. **拟合形如** $f(x)\approx\dfrac{a+bx}{1+cx}$ **的函数的一种快速方法是将最小二乘法用于下列问题：** $f(x)(1+cx)\approx a+bx$**，试用这一方法拟合下面给出的世界人口数据：**

年	人口(百万)
1000	340
1650	545
1800	907
1900	1610
1950	2509
1970	3650

要求用 MATLAB **语句写出上述最小二乘法确定系数** a,b,c **的程序.**

习题答案

习题一

1. **构造性证明**:由 $\cos(2\cos^{-1}x)=\frac{1}{2}$ 得 $x=\pm\frac{\sqrt{3}}{2}$.

非构造性证明:令 $f(x)=2\cos(2\cos^{-1}x)-1$,由连续函数零点定理得在 $[0,1]$ 和 $[-1,0]$ 上分别至少各有一零点.

2. $I=\arctan\frac{1}{1+N+N^2}$.

3. (4)最好.

4. $e_a(x_1\pm x_2)\approx e_a(x_1)\pm e_a(x_2)$,

$e_r(x_1\pm x_2)\approx\frac{x_1^*}{x_1^*\pm x_2^*}e_r(x_1)\pm\frac{x_2^*}{x_1^*\pm x_2^*}e_r(x_2)$,

$e_a(x_1x_2)\approx x_2^*e_a(x_1)+x_1^*e_a(x_2)$, $e_r(x_1x_2)\approx e_r(x_1)+e_r(x_2)$.

5. 不稳定,误差 $\approx\frac{1}{2}\times10^8$.

6. $x_1\approx55.98$, $x_2\approx0.01786$.

习题二

1. $P_2(x)=\frac{1}{2}x^2+\frac{1}{2}x+1$.

2. 取 $x_0=0.5$, $x_1=0.6$,用线性插值得 $\sin0.56891\approx0.54667$.

取 $x_0=0.5$, $x_1=0.6$, $x_2=0.7$,用二次插值得 $\sin0.56891\approx0.54714$.

$|R_2(x)|\leqslant2.95\times10^{-5}$.

3. 取 $f(x)=x^k$,则 $L_n(x)=\sum_{i=0}^{n}x_i^k l_i(x)\equiv f(x)=x^k$.

4. $h=0.002$.

5. $f[2^0,2^1,\cdots,2^7]=1$, $f[2^0,2^1,\cdots,2^8]=0$.

6. $\ln1.05\approx0.04879072$.

7. $N_5(x)=N_5(0.0+0.1t)=0.02t^2+0.30t+1.00$, $(0\leqslant t\leqslant1)$.

8. $P(x)=4x^3-3x$, $R(x)=\frac{1}{4!}f^{(4)}(\xi)x^2(x-1)^2$.

9. $H(x)=\frac{x^2}{4}(x-3)^2$, $R(x)=\frac{1}{5!}f^{(5)}(\xi)x^2(x-1)^2(x-2)$.

10. $S(x)=\begin{cases}\frac{9}{4}x^3-\frac{27}{4}x^2+\frac{5}{2}x, & 0\leqslant x<1; \\ \frac{1}{4}x^3-\frac{3}{4}x^2-\frac{7}{2}x+2, & 1\leqslant x<4; \\ -\frac{3}{4}x^3+\frac{45}{4}x^2-\frac{103}{2}x+66, & 4\leqslant x\leqslant 5.\end{cases}$

11. 由已知

x_0	x_1
$f(x_0)$	$f(x_1)$
$f'(x_0)$	

用待定系数法构造 $H_2(x)$，再证其余项

$$R(x)=\frac{1}{6}(x-x_0)^2(x-x_1)f'''(\xi).$$

12. (1) $T_m[T_n(x)]=\cos mn\theta=T_{mn}(x)$.

(2) $2T_m(x)T_n(x)=2\cos m\theta\cdot\cos n\theta=\cos(m+n)\theta+\cos(m-n)\theta$

$$=T_{m+n}(x)+T_{m-n}(x).$$

13. $S_1^*(x)=0.42696x+0.93432$.

14. $S_2^*(x)=-4.12251x^2+4.12251x-0.050465$.

15. 用一次多项式 $y=a_0+a_1x$ 拟合的法方程为

$$\begin{cases}9a_0=18.1183\\3.75a_1=8.4437\end{cases};$$

用二次多项式 $y=a_0+a_1x+a_2x^2$ 拟合的法方程为

$$\begin{cases}9a_0+3.75a_2=18.1183\\3.75a_1=8.4437\\3.75a_0+2.7656a_2=7.58696\end{cases};$$

用三次多项式 $y=a_0+a_1x+a_2x^2+a_3x^3$ 拟合的法方程为

$$\begin{cases}9a_0+3.75a_2=18.1183\\3.75a_1+2.7656a_3=8.4437\\3.75a_0+2.7656a_2=7.5869\\2.7656a_1+2.3877a_3=6.2820\end{cases}.$$

16. $I_0=e^{\alpha_0}=e^{1.73}\approx 5.64$，$\alpha\approx 2.89$，$I=5.46e^{-2.89t}$.

习题三

1. (1) $T_8\approx 0.1114024$，$S_4\approx 0.111572$

(2) $T_4\approx 17.22774$，$S_2\approx 17.32223$.

2. 因 $R_T=\frac{1}{12}h^2|f''(\eta)|\leqslant\frac{h^2}{12}\leqslant\frac{1}{2}\times10^{-4}$，即 $n^2\geqslant\frac{1}{6}\times10^4$，所以用复化梯形公式求积分，需将[0,1]分成 41 等分，而用复化辛甫生公式求积，因 $R_s=\frac{h^4}{180}|f^{(4)}(\eta)|\leqslant\frac{h^4}{180}\leqslant\frac{1}{2}\times10^{-4}$，只需将[0,1]分为 4 等分.

3. 当 $f''(x)>0$ 时，曲线 $y=f(x)$ 向上凹.

4. 依次令 $f(x)=1,x,x^2,x^3$ 代入，求积公式精确成立，而用 $f(x)=x^4$ 代入，求积公式不精确成立.

5. $\int_a^b f(x)\mathrm{d}x\approx(b-a)\left[\frac{1}{8}f(a)+\frac{3}{8}f(a+h)+\frac{3}{8}f(a+2h)+\frac{1}{8}f(b)\right]$具有三次代数精度. $\int_0^1\frac{1}{1+x^2}\mathrm{d}x\approx0.7846$.

6. 利用定积分定义可证.

7. (1) $\omega_0=\frac{1}{3}$, $\omega_1=\frac{4}{3}$, $\omega_2=\frac{1}{3}$；三次代数精度.

(2) $\omega_{-1}=\frac{8}{3}h$, $\omega_0=-\frac{4}{3}h$, $\omega_1=\frac{8}{3}h$；三次代数精度.

(3) $\omega_1=\frac{2}{3}h$, $\omega_2=\frac{1}{3}h$, $\omega_3=\frac{1}{6}h^2$；二次代数精度.

(4) $a=\frac{1}{12}$，三次代数精度.

8. $R_1=-12.070419$.

9. (1) $J\approx1.0986125$；

(2) 用三点高斯公式：$J=\int_{-1}^1\frac{\mathrm{d}t}{2+t}\approx\sum_{i=0}^2\omega_i f_i\approx1.0980393$；

用五点高斯公式：$J=\int_{-1}^1\frac{\mathrm{d}t}{2+t}\approx\sum_{i=0}^4\omega_i f_i\approx1.098609247$；

(3) $J=\sum_{k=1}^4 J_k=\int_1^{1.5}+\int_{1.5}^2+\int_2^{2.5}+\int_{2.5}^3\approx1.098537574$.

10. $f'(1.0)\approx-0.2470$, $f'(1.1)\approx-0.2170$, $f'(1.2)\approx-0.1870$.

习题四

1. $x\approx1.921875$. 误差为 $\frac{1}{2^6}$.

2. 迭代格式(1),(2)收敛，迭代格式(3)发散. 若选格式(1)迭代 12 次得 $x\approx1.4657170$.

3. $x \approx 0.739$.

4. $x_{k+1} = \dfrac{2}{3} x_k + \dfrac{a}{3x_k^2}$.

5. $x \approx 1.8794$.

习题五

1. (1) **由消去过程得**

$$[\boldsymbol{A} \vdots \boldsymbol{b}] \longrightarrow \left[\begin{array}{cccc:c} 1 & 2 & 1 & -2 & 4 \\ 0 & 1 & 1 & 2 & -1 \\ 0 & 0 & 3 & -3 & 9 \\ 0 & 0 & 0 & 3 & -3 \end{array}\right]$$

由回代过程得：$x_4 = -1, x_3 = 2, x_2 = -1, x_1 = 2$；

(2) $\boldsymbol{A} \longrightarrow \begin{bmatrix} 1 & 2 & 1 & -2 \\ 2 & 1 & 1 & 2 \\ -2 & 2 & 3 & -3 \\ 1 & 1 & 0 & 3 \end{bmatrix}$， **求解** $\boldsymbol{LY} = \boldsymbol{b}$ **得** $\boldsymbol{Y} = (4, -1, 9, -3)^{\mathrm{T}}$

求解 $\boldsymbol{UX} = \boldsymbol{Y}$ **得** $\boldsymbol{X} = (2, -1, 2, -1)^{\mathrm{T}}$.

2. $\boldsymbol{X} = (2, 3, 2, 1)^{\mathrm{T}}$.

3. (1) $\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$， $\boldsymbol{Y} = \left(0, 3, \dfrac{1}{2}\right)^{\mathrm{T}}$， $\boldsymbol{X} = (1, -1, 1)^{\mathrm{T}}$；

(2) $\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 1 & 7 & 6 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 2 & 6 & 12 \\ 0 & 0 & 6 & 24 \\ 0 & 0 & 0 & 24 \end{bmatrix}$， $\boldsymbol{Y} = (2, 8, 18, 24)^{\mathrm{T}}$， $\boldsymbol{X} = (-1, 1, -1, 1)^{\mathrm{T}}$.

4. **利用矩阵乘法，类同于教材中的杜利脱尔分解的推导**.

5. **用线性代数知识证（略）**.

6. $\boldsymbol{L} = \begin{bmatrix} 1 & & & \\ 1 & 2 & & \\ 1 & 1 & 3 & \\ 1 & 1 & 1 & 4 \end{bmatrix}$.

7. $\boldsymbol{L} = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 4 & 0 \\ -2 & 2 & 1 \end{bmatrix}$， $\boldsymbol{Y} = (5, 2, -1)^{\mathrm{T}}$

$\boldsymbol{X} = (2, 1, -1)^{\mathrm{T}}$.

8. $\boldsymbol{A}=\begin{bmatrix} 2 & & & \\ -1 & \frac{3}{2} & & \\ & -1 & \frac{4}{3} & \\ & & -1 & \frac{5}{4} \end{bmatrix}\begin{bmatrix} 1 & -\frac{1}{2} & & \\ & 1 & -\frac{2}{3} & \\ & & 1 & -\frac{3}{4} \\ & & & 1 \end{bmatrix}$

$\boldsymbol{Y}=\left(\frac{1}{2},\frac{2}{3},\frac{3}{4},\frac{4}{5}\right)^{\mathrm{T}}$，$\boldsymbol{X}=\left(\frac{77}{60},\frac{47}{30},\frac{27}{20},\frac{4}{5}\right)^{\mathrm{T}}$.

9. (1) $\mathrm{Cond}(\boldsymbol{A})_{\infty}=289$；

(2) $\frac{\|\delta\boldsymbol{X}\|_{\infty}}{\|\boldsymbol{X}\|_{\infty}}=289\times\frac{0.01}{1}=2.89$.

10. (1) J-**迭代分量形式**：

$$\begin{cases} x_1^{(m+1)}=1.2-0.1x_2^{(m)}-0.15x_3^{(m)} \\ x_2^{(m+1)}=1.5-0.125x_1^{(m)}-0.125x_3^{(m)} \\ x_3^{(m+1)}=2-\frac{2}{15}x_1^{(m)}+0.2x_2^{(m)} \\ \qquad (m=0,1,2,\cdots) \end{cases};$$

矩阵形式：

$$\begin{bmatrix} x_1^{(m+1)} \\ x_2^{(m+1)} \\ x_3^{(m+1)} \end{bmatrix}=\begin{bmatrix} 0 & -\frac{1}{10} & -\frac{3}{20} \\ -\frac{1}{8} & 0 & -\frac{1}{8} \\ -\frac{2}{15} & \frac{1}{5} & 0 \end{bmatrix}\begin{bmatrix} x_1^{(m)} \\ x_2^{(m)} \\ x_3^{(m)} \end{bmatrix}+\begin{bmatrix} \frac{6}{5} \\ \frac{3}{2} \\ 2 \end{bmatrix};$$

迭代七次得近似值 $\boldsymbol{X}\approx(0.76735,1.13841,2.12537)^{\mathrm{T}}$.

(2) **用赛德尔迭代法迭代的分量形式**：

$$\begin{cases} x_1^{(m+1)}=1.2-0.1x_2^{(m)}-0.15x_3^{(m)} \\ x_2^{(m+1)}=1.5-0.125x_1^{(m+1)}-0.125x_3^{(m)} \\ x_3^{(m+1)}=2-\frac{2}{15}x_1^{(m+1)}+0.2x_2^{(m+1)}, \quad (m=0,1,2,\cdots) \end{cases};$$

矩阵形式：

$$\begin{bmatrix} x_1^{(m+1)} \\ x_2^{(m+1)} \\ x_3^{(m+1)} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{10} & -\frac{3}{20} \\ 0 & \frac{1}{80} & -\frac{17}{160} \\ 0 & \frac{19}{1200} & -\frac{1}{800} \end{bmatrix} \begin{bmatrix} x_1^{(m)} \\ x_2^{(m)} \\ x_3^{(m)} \end{bmatrix} + \begin{bmatrix} \frac{6}{5} \\ \frac{27}{20} \\ \frac{211}{100} \end{bmatrix};$$

迭代五次得近似值 $\boldsymbol{X} \approx (0.76735, 1.13841, 2.12537)^{\mathrm{T}}$.

11. 由 $\rho(\boldsymbol{B}_{\mathrm{J}})=0<1$，得雅可比迭代收敛；而 $\rho(\boldsymbol{B}_{\mathrm{S}})=2>1$，所以 G-S 迭代发散.

12. 由 $\rho(\boldsymbol{B}_{\mathrm{J}})=1$ 得雅可比迭代发散；又由 $\boldsymbol{A}$ 为对称正定，所以 G-S 迭代与 SOR 迭代（当 $0<\omega<2$ 时）均收敛.

13. (1) 由 $\rho(\boldsymbol{B}_{\mathrm{S}})=\sqrt{\frac{1}{24}}<1$，得 G-S 迭代收敛；

(2) 由 $\rho(\boldsymbol{B}_{\mathrm{S}})=3\frac{1}{8}$，得 G-S 迭代发散.

14. 迭代 8 次得 $\boldsymbol{X} \approx (-4.000025, 2.999988, 2.000003)^{\mathrm{T}}$.

15. 显然不满足[判别条件 I]，但因 $\rho(\boldsymbol{B}_{\mathrm{J}})<1$，所以雅可比迭代收敛.

习题六

1. 欧拉方法 $\begin{cases} y_{n+1}=y_n+h(-y_n+x_n+1) \\ y(0)=1, \quad (n=0,1,2,\cdots) \end{cases}$

x_n	0.0	0.1	0.2	0.3	0.4	0.5
y_n	1.000000	1.000000	1.010000	1.029000	1.056100	1.090490

x_n	0.6	0.7	0.8	0.9	1.0
y_n	1.131441	1.178297	1.230467	1.287420	1.348678

2.

x_n	0	0.2	0.4	0.6	0.8	1.0
y_n	1	1.2428	1.583636	2.044213	2.651042	3.436502
$y(x_n)$	1	1.242806	1.583649	2.044238	2.651082	3.436564

3. **按二阶方法定义证**

$$|\psi(x,y,h)-\psi(x,y^*,h)|\leqslant L_0\left(1+\frac{h}{2}L_0\right)|y-y^*|$$

即得所证 L **表示式.**

4. $|1-10h+50h^2|\leqslant 1$,**即** $0<h\leqslant 0.2$.

5. **设** $y_n=y(x_n)$,**由公式定义的** y_{n+1} **和** $y(x_{n+1})$**的泰勒展开式比较,可证得** $y_{n+1}-y(x_{n+1})=O(h^3)$.

6. $a=1,b=\frac{1}{3},c=\frac{4}{3},d=\frac{1}{3}$.

实习题答案

实习题一

1. $R\approx 0.785397$，2. $R\approx -12.070419$，3. $R\approx 440.536$.

实习题二

1. **两种方法的前五次迭代结果列于下表：**

高斯-赛德尔迭代和 SOR 迭代计算结果

	高斯-赛德尔迭代($\omega=1$)			SOR 迭代($\omega=1.25$)		
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	1	1	1	1	1	1
1	5.250000	3.812500	−5.046875	6.312500	3.5195313	−6.6501465
2	3.1406250	3.8828125	−5.0292969	2.6223145	3.9585266	−4.6004238
3	3.0878906	3.9267587	−5.0183105	3.1333027	4.0102646	−5.0966863
4	3.0549316	3.9542236	−5.011441	2.9570512	4.0074838	−4.9734897
5	3.0343323	3.9713898	−5.0071526	3.0037211	4.0029250	−5.0057135

进一步计算表明，若要得到七位有效数字的近似解 $x^{(k)}$，用 G-S 迭代需要迭代 34 次，而用 $\omega=1.25$ 的 SOR 迭代只需 14 次，可见，适当地选择松驰因子 ω，可以大大加快收敛速度.

2. **前十次迭代结果列于下表：**

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$x_5^{(k)}$	$x_6^{(k)}$	$x_7^{(k)}$
1	.66532	−1.65487	3.38602	−2.84311	3.71733	−6.47660	6.33758
2	.45488	−1.64243	2.68754	−3.95407	4.73673	−5.91957	6.55884
3	1.06912	−1.99307	3.02628	−3.99170	4.96120	−6.04299	6.53782
4	.96886	−2.03696	3.01162	−4.03571	4.95828	−6.05742	6.52570
5	.98180	−2.04553	3.01707	−4.03920	4.96252	−6.06219	6.52447
6	.98065	−2.04678	3.01694	−4.04055	4.96278	−6.06274	6.52416

续表

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$x_5^{(k)}$	$x_6^{(k)}$	$x_7^{(k)}$
7	.980 98	−2.047 09	3.017 10	−4.040 72	4.962 92	−6.062 90	6.524 12
8	.980 96	−2.047 14	3.017 10	−4.040 76	4.962 94	−6.062 94	6.524 11
9	.980 97	−2.047 15	3.017 11	−4.040 77	4.962 94	−6.062 93	6.524 10
10	.980 97	−2.047 15	3.017 11	−4.040 77	4.962 94	−6.062 93	6.524 10

[**近似解**]　**取** $\boldsymbol{X}=\boldsymbol{X}^{(10)}=(x_1,x_2,\cdots,x_7)^{\mathrm{T}}$.

其中　$x_1\approx 0.980\,97$，$x_2\approx -2.047\,15$，$x_3\approx 3.017\,11$，

$x_4\approx -4.040\,77$，$x_5\approx 4.962\,94$，$x_6\approx -6.062\,93$，

$x_7\approx 6.524\,10$.

实习题三

1.

n	0	1	2	3	4	5
x_n	0	0.2	0.4	0.6	0.8	1.0
y_n	1	1.183 23	1.341 67	1.483 28	1.612 51	1.732 14

2.

n	x_n	y_n	n	x_n	y_n
0	0	1	6	0.6	1.148 811 636 0
1	0.1	1.004 837 410 8	7	0.7	1.196 585 303 8
2	0.2	1.018 730 753 1	8	0.8	1.249 328 964 1
3	0.3	1.048 018 220 7	9	0.9	1.306 569 659 7
4	0.4	1.070 320 046 0	10	1.0	1.367 879 441 2
5	0.5	1.106 530 649 7			

实习题四

1. $\boldsymbol{X}$=[0.25,0.50,0.75,1.00,1.25,1.50,1.75,2.00,
2.25,2.50]′;

A=[exp(x),sin(x),log(x),cos(x)];

Y=[1.284,1.648,2.117,2.718,3.427,2.798,3.534,
4.456,5.465,5.894]′

$$C=A\backslash Y=\begin{pmatrix}0.4815\\1.0057\\0.2731\\0.7740\end{pmatrix}=\begin{pmatrix}a\\b\\c\\d\end{pmatrix}.$$

2. X=[−0.25,0.00,0.25,0.5]′;

Y=[1.439,2.003,2.564,3.133]′;

y_0=interpl(X,Y,0.13,′cubic′)

y_0=2.2944.

3. **先建立** Len.m **文件**

function f=len(t)

f=atan(t)./t.Λ2;

再在主窗口中;

I=quad(′len′,0.1,0.5)

I=1.5722.

4. **先建立** S.m **文件**

function g=S(x) g=X.Λ9−522+exp(x);

再在主窗口中:x_0=f zero(′s′,1.9);X_0=2.0011.

5. **先建立** ode.m **文件**:

function dy=ode(x,y)

dy=sqrt(x.Λ2+y.Λ2)

再在主窗口中:

[x,y]=ode45(′ode′,[0∶0.01∶1.0],−1)

得最后 8 **个点上的函数值**:

x	0.9300	0.9400	0.9500	0.9600	0.9700	0.9800	0.9900	1.0000
y	−0.2211	−0.2115	−0.2018	−0.1921	−0.1822	−0.1723	−0.1623	−0.1523

6. **原理**:**利用一般迭代法的矩阵形式**:

$$x^{(m+1)}=B_J X^{(m)}+g$$

若其迭代方阵 B_J **的谱半径** $\rho(B_J)<1$，**则迭代收敛**.

```
a=hi(b(100)+0.5speye(100));
l=-1.*tril(a,-1);
u==-1.*triu(a,1);
d=a+l+u;
bj=inv(d)*(l+u);
w=eig(bj);
max(w)
ans=0.4508,
```

即 $\rho(B_J)<1$，**所以迭代收敛**.

7.
```
function  X=bidiag(N,a,d,b);
k=(N+1)/2;x(1)=b(1)/d(i);
for i=2:(k-1)
X(i)=(b(i)-a(i-1)*X(i-1))/d(i);
end
X(N)=b(N)/d(N);
for i=N-1:(-1):(k+1)
X(i)=(b(i)-a(i)*X(i+1))/d(i);
end
X(k)=(b(k)-a(k-1)*X(k-1)-a(k)*Xk+1))/d(k);
```

8. **由于** $f(x)(1+cx)\approx a+bx \Leftrightarrow a+bx-cf(x)x\approx f(x)$，**故** MATLAB **程序如下**：

```
X=[1000  1650  1800  1900  1950  1970]';
Y=[340  545  907  1610  2509  3650]';
Z=ones(length(x),1);
u=X.*Y;
v=(-1)*u;
A=[z  x  v];
W=A\Y
```

运行结果：a=w(1)，b=w(2)，c=w(3)

参考书目

[1] 李庆扬,易大义,王能超编. 现代数值分析. 北京:高等教育出版社,1995

[2] 武汉大学,山东大学计算数学教研室. 计算方法. 北京:人民教育出版社,1979

[3] 杨凤翔,陆君良编. 数值分析. 天津:天津大学出版社,1985

[4] 易大义,蒋叔豪,李有法编著. 数值方法. 杭州:浙江科学技术出版社,1987

[5] 沈剑华编. 计算数学基础. 上海:同济大学出版社,1989

[6] K·E·阿特金森. 数值分析引论(中译本). 上海:上海科学技术出版社,1986

[7] 冯康等. 数值计算方法. 北京:国防工业出版社,1978

[8] 同济大学计算数学教研室编. 数值分析基础. 上海:同济大学出版社,1998

[9] 同济大学计算数学教研室编. 数值计算解题方法与同步训练. 上海:同济大学出版社,2001